

AD 696567

NPS-58DW9071A

# UNITED STATES NAVAL POSTGRADUATE SCHOOL



NOV 1969

DIGITAL ANALYSIS OF TURBULENCE DATA  
ON THE IBM 360/67 AT THE  
NAVAL POSTGRADUATE SCHOOL

by

J. R. Wilson  
N. E. J. Boston  
W. W. Denner

1 July 1969

This document has been approved for public  
release and sale; its distribution is unlimited.

Reproduced by the  
CLEARINGHOUSE  
for Federal Scientific & Technical  
Information Springfield Va 22151

100

NAVAL POSTGRADUATE SCHOOL  
Monterey, California

Rear Admiral Robert W. McNitt, USN  
Superintendent

R. F. Rinehart  
Academic Dean

ABSTRACT:

A system of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.

The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FCPLLOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra phase and coherence are also printed out. The program UBC FCPLLOT provides a Calcomp plot of the spectra for qualitative analysis.

These programs have been tested on the IBM 360/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U.B.C. machine.

A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.

This task was supported by: Naval Ordnance Systems Command  
Navy Department

Noël E. J. Boston

N. E. J. Boston  
Assistant Professor of Oceanography

Approved by:

ORIGINAL SIGNED BY  
DAVID F. LEIPPER

D. F. Leipper  
Chairman  
Department of Oceanography

Warren W. Denner

W. W. Denner  
Assistant Professor of Oceanography

Released by:

C. E. Menneken

C. E. Menneken  
Dean of Research Administration

NPS-58DW9071A

NAVAL POSTGRADUATE SCHOOL

Department of Oceanography  
Monterey, California

DIGITAL ANALYSIS OF TURBULENCE DATA ON THE IBM 360/67  
AT THE NAVAL POSTGRADUATE SCHOOL

by

J. R. Wilson

N. E. J. Boston

W. W. Denner

This research was sponsored by Naval Ordnance Systems Command under NAVORD CONTRACT ORD TASK-03C-005/551-1/UR104-03-01. The work reported herein is of a preliminary nature and the results are not necessarily in final form. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

July 1969

## FORWARD

The digital programs discussed in this report were developed at the University of British Columbia by John Garrett and Ron Wilson during 1967. They have been applied to a large variety of geophysical turbulence data and found extremely successful. Because of the nature of the air/sea interaction program being carried out by the Department of Oceanography of the Naval Postgraduate School, such programs were desirable to be available at the computing facilities of the School.

In February of 1969, Mr. Ron Wilson was hired as a consultant to convert the University of British Columbia programs for use on the IBM 360/67 computer of the Naval Postgraduate School. This report is a summary of the system based on notes provided by Mr. Wilson. The principal investigators take full responsibility for any errors or omissions that may occur in this report.

Noël Boston  
W. W. Denner  
Principal Investigators

## ABSTRACT

A system of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.

The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FCPLLOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra, phase and coherence are also printed out. The program UBC FCPLLOT provides a Calcomp plot of the spectra for qualitative analysis.

These programs have been tested on the IBM 360/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U. B. C. machine.

A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.

## CONTENTS

	Page
FORWARD	3
ABSTRACT	4
LIST OF FIGURES	6
LIST OF TABLES	7
I. GENERAL	8
II. DIGITAL SAMPLING PROCEDURE	9
A. Analog Signal Level	9
B. Aliasing	9
C. Sampling Rate	10
D. Miscellaneous Factors	11
III. COMPUTATION OF SPECTRA AND CROSS SPECTRA	12
IV. INPUT DATA TAPE FORMAT	25
V. PROCEDURE IN ANALYZING DATA	27
VI. SOURCE PROGRAMS	28
ACKNOWLEDGEMENTS	29
REFERENCES	30
APPENDIX I ANALYZING DATA DIGITIZED ON THE SDS-9300	31
APPENDIX II FORTRAN SOURCE PROGRAMS	33

# LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Digital Sampling Procedure	10
2	FCPLOT of DISA U Spectrum	20
3	FCPLOT of Sonic W Spectrum	21
4	FCPLOT of UW Co-spectrum	22
5	FCPLOT of UW Quadrature Spectrum	23
6	FCPLOT of Fourier Coefficients for DISA U Data	24

# LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
I	DISA U Spectrum	15
II	Sonic W Spectrum	16
III	UW Co-spectrum	17
IV	UW Quadrature Spectrum	18
V	Fourier Coefficients for DISA U data	19

## I. GENERAL

This report is intended (a) to summarize the system of time series programs used by the Institute of Oceanography of the University of British Columbia which are now available to the Department of Oceanography of the Naval Postgraduate School and (b) to describe briefly the procedures to be followed in using them.

These programs were made available in February 1969 and this report describes the programs as they existed at that time. Some updating and expansion of these programs may occur from time to time. A check on their current state is recommended prior to using.

The turbulence data used to test these programs included downstream velocity fluctuations measured with a hot-wire anemometer (DISA U) and vertical velocity fluctuations measured with a sonic anemometer (Sonic W). These measurements were made simultaneously during August 1968 at the U.B.C. Institute of Oceanography experimental site at Spanish Banks, British Columbia.

## II DIGITAL SAMPLING PROCEDURE

The system is based on the assumption that the data to be analyzed are initially recorded on analog tape. These analog data must then be converted to digital form before they can be processed by a digital computer. The analog to digital procedure is of central importance and must be done with extreme care. Almost always there must be some conditioning of the signal prior to the analog-to-digital (A/D) conversion. Common problems to consider are analog signal level, aliasing and sampling frequency. Of course each analysis problem presents its unique difficulties and must be treated individually.

### A. Analog Signal Level

The patch of signal to be analyzed may not have been recorded under ideal conditions. It may vary from almost too small (approaching system noise level) to almost too large (occasional clipping). To insure good digitizing some amplification may be necessary to provide the optimum signal input to the A/D converter. If there is an appreciable DC level on the signal, high pass filtering or rerecording may be necessary before digitization can begin. In brief, the signal level must always be checked to be sure maximum advantage is being taken of the full dynamic range of the A/D converter.

### B. Aliasing

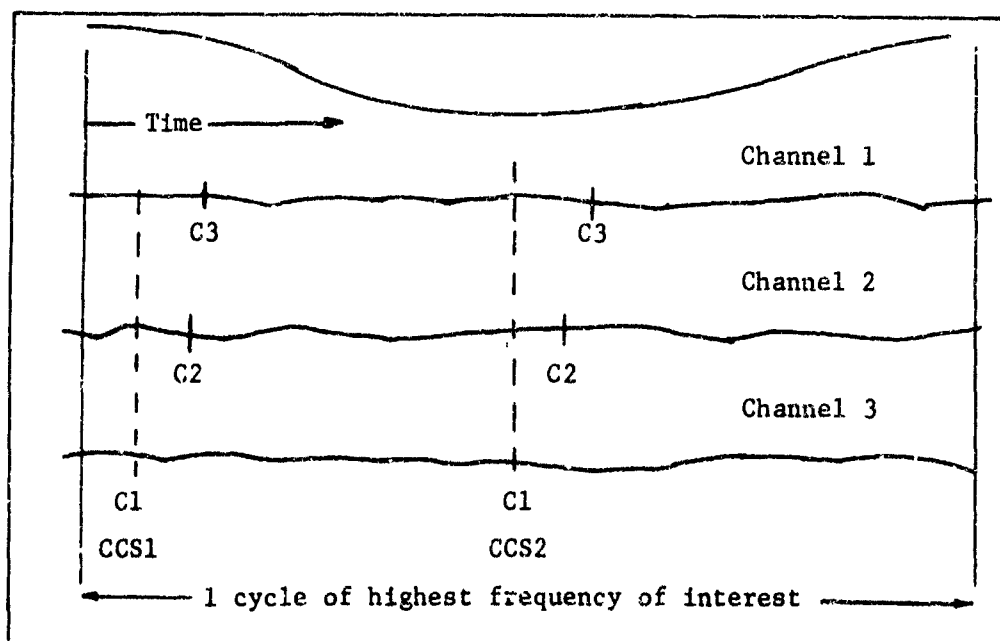
When an analog signal is sampled at a frequency  $F$ , energy present in the signal at frequencies  $f > \frac{F}{2}$  will appear in the sampled output at an apparent (aliased) frequency  $F-f$ . This phenomenon is known as aliasing

1000 1000 1000

1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75  
 76  
 77  
 78  
 79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97  
 98  
 99  
 100  
 101  
 102  
 103  
 104  
 105  
 106  
 107  
 108  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117  
 118  
 119  
 120  
 121  
 122  
 123  
 124  
 125  
 126  
 127  
 128  
 129  
 130  
 131  
 132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152  
 153  
 154  
 155  
 156  
 157  
 158  
 159  
 160  
 161  
 162  
 163  
 164  
 165  
 166  
 167  
 168  
 169  
 170  
 171  
 172  
 173  
 174  
 175  
 176  
 177  
 178  
 179  
 180  
 181  
 182  
 183  
 184  
 185  
 186  
 187  
 188  
 189  
 190  
 191  
 192  
 193  
 194  
 195  
 196  
 197  
 198  
 199  
 200  
 201  
 202  
 203  
 204  
 205  
 206  
 207  
 208  
 209  
 210  
 211  
 212  
 213  
 214  
 215  
 216  
 217  
 218  
 219  
 220  
 221  
 222  
 223  
 224  
 225  
 226  
 227  
 228  
 229  
 230  
 231  
 232  
 233  
 234  
 235  
 236  
 237  
 238  
 239  
 240  
 241  
 242  
 243  
 244  
 245  
 246  
 247  
 248  
 249  
 250  
 251  
 252  
 253  
 254  
 255  
 256  
 257  
 258  
 259  
 260  
 261  
 262  
 263  
 264  
 265  
 266  
 267  
 268  
 269  
 270  
 271  
 272  
 273  
 274  
 275  
 276  
 277  
 278  
 279  
 280  
 281  
 282  
 283  
 284  
 285  
 286  
 287  
 288  
 289  
 290  
 291  
 292  
 293  
 294  
 295  
 296  
 297  
 298  
 299  
 300  
 301  
 302  
 303  
 304  
 305  
 306  
 307  
 308  
 309  
 310  
 311  
 312  
 313  
 314  
 315  
 316  
 317  
 318  
 319  
 320  
 321  
 322  
 323  
 324  
 325  
 326  
 327  
 328  
 329  
 330  
 331  
 332  
 333  
 334  
 335  
 336  
 337  
 338  
 339  
 340  
 341  
 342  
 343  
 344  
 345  
 346  
 347  
 348  
 349  
 350  
 351  
 352  
 353  
 354  
 355  
 356  
 357  
 358  
 359  
 360  
 361  
 362  
 363  
 364  
 365  
 366  
 367  
 368  
 369  
 370  
 371  
 372  
 373  
 374  
 375  
 376  
 377  
 378  
 379  
 380  
 381  
 382  
 383  
 384  
 385  
 386  
 387  
 388  
 389  
 390  
 391  
 392  
 393  
 394  
 395  
 396  
 397  
 398  
 399  
 400  
 401  
 402  
 403  
 404  
 405  
 406  
 407  
 408  
 409  
 410  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447  
 448  
 449  
 450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470  
 471  
 472  
 473  
 474  
 475  
 476  
 477  
 478  
 479  
 480  
 481  
 482  
 483  
 484  
 485  
 486  
 487  
 488  
 489  
 490  
 491  
 492  
 493  
 494  
 495  
 496  
 497  
 498  
 499  
 500  
 501  
 502  
 503  
 504  
 505  
 506  
 507  
 508  
 509  
 510  
 511  
 512  
 513  
 514  
 515  
 516  
 517  
 518  
 519  
 520  
 521  
 522  
 523  
 524  
 525

### C. Sampling Rate

If three channels are to be sampled and co-spectra are required between all combinations the sampling procedure might be as illustrated in Figure 1.



### Figure 1. Sampling Procedure

The cuts C1, C2 and C3 represent the digital samplings of channels 1, 2 and 3 respectively.

CCS1 and CCS2 represent two samplings of all channels and are referred to as Cross Channel Sweeps.

The time lag between sampling channel 1 and channel 3 in a cross channel sweep should be a small fraction of a cycle since the program assumes they are sampled simultaneously for purposes of computing the co-spectra and quadrature spectra.

The time between cross channel sweeps should be such that each channel is sampled at least twice per cycle of the highest frequency with sufficient amplitude to produce an aliasing problem.

#### D. Miscellaneous Factors

It is important that all factors influencing the validity and interpretations of results be considered prior to digitizing. Besides those factors already discussed these should include the record length, the block length, spectral windows and how these will effect the stability of the spectral estimates. Further, an estimate of the computer time required for the analysis should be made. There has not been sufficient experience with the computing facilities at the Naval Postgraduate School to allow tables of time for typical calculations to be presented. However the tables given by Garrett (1967) will indicate order of magnitude computing times. In addition his report contains excellent discussions of some of the factors mentioned here.

### III COMPUTATION OF SPECTRA AND CROSS SPECTRA

The computation of the spectral densities employ the Fast Fourier transform (FFT) of Cooley and Tukey (1965). The program used is distributed through the share library and is catalogued at PKFORT SDA3465.

The subroutine will transform a number of data points equal to  $2^n$  where  $n$  is a positive integer less than or equal to 13. This means the largest number of data points that can be transformed is 8192. Since the time series encountered in turbulence measurements generate more points than this, some method of transforming and averaging consecutive blocks of data must be used. This also makes it possible to gain some information as to the variance of the spectral or cospectral densities over the interval of the signal.

The method employed consists of evaluating the spectrum, co-spectrum or quadrature spectrum for each frequency band for successive blocks of  $2^n$  data points. The mean and variance of these densities are then formed and printed and plotted.

For example purposes, assume 100 blocks of 2048 points each are transformed. For each block there will be contained in the Fourier coefficients an amplitude and phase for frequencies corresponding to 0, 1, 2, ..., 1024 cycles over the 2048 points. Zero is the mean and 1024 is the folding frequency. Therefore for each block of 2048 data points there are 1024 estimates of spectral, cospectral or quadrature spectral density. This is considered to be too much information to plot graphically so in the production of printouts and plots further averaging than along block must be done. For example the program may divide the spectrum into 32 bands and average across these bands as well as along blocks to obtain the mean and variance of the quantities.

If there are two channels and they have for the  $i^{\text{th}}$  harmonic, amplitudes  $A_i$  and  $B_i$ , and phases  $\alpha_i$  and  $\beta_i$ , then the spectral densities are

$$\frac{A_i^2}{2 \times \text{Bandwidth}} \quad \text{and} \quad \frac{B_i^2}{2 \times \text{Bandwidth}}$$

where the bandwidth is given by

$$\frac{\text{sampling frequency}}{\text{no. of data points in the transform}}$$

The cospectrum is given by

$$\frac{A_i \times B_i \cos(\alpha_i - \beta_i)}{2 \times \text{Bandwidth}}$$

The quadrature spectrum is given by

$$\frac{A_i \times B_i \sin(\alpha_i - \beta_i)}{2 \times \text{Bandwidth}}$$

These quantities are computed for each frequency and block and are then averaged across the frequencies and along blocks.

The error bars plotted are computed by evaluating the variance or the estimates of the spectral density in the band and dividing by the square root of the number of estimates to establish the expected variance in the mean. In the case of the SCOR program the plotted and printed error bars represent standard deviations in the mean. The FCPLOT program uses 1.96 times the standard deviation (or some other kind of 95% confidence interval).

The values of phase and coherence produced by the SCOR program are derived directly from the final values of the co- and quadrature spectra appearing to their left (Table II, for example).

Further information is available on the comments cards with the Fortran source programs (Appendix II).

Examples of print out for spectra, co-spectrum, quadrature spectrum and Fourier coefficients are given in Tables I through V. The corresponding FCPLTs for these tables are Figures 2 through 6.



TABLE II

SPECTRUM STATISTICS FOR 102 70028 CHANNELS 2 AMPERES EACH TAPE 140/1/10/9/68  
 THE SAMPLING FREQUENCY WAS 62.77105AMP/SEC. MAKING THE BLOCK LENGTH 16.31326 SECONDS  
 TREND IS THE AVERAGE OF (VALUE(AI)-VALUE(R))/BLOCK NO. (AI)-BLOCK NO. (R))  
 A CALIBRATION FACTOR OF 6.540E-01 HAS BEEN APPLIED TO THE INPUT DATA

FREQUENCY HERTZ	BANDWIDTH	SPECTRUM (M/SEC	STD. DEV. 1#2/HERTZ	TREND	FREQ*SPECTRUM (M/SEC	LAST HARMON
1 31E-02	6.13E-02	8.25E-02	1.17E-01	-8.084E-04	4.380E-03	1
2 1.9E-01	6.13E-02	6.19E-02	8.00E-02	-2.143E-04	7.352E-03	2
3 2.43E-01	6.13E-02	5.14E-02	5.88E-02	-5.578E-05	1.102E-03	3
4 3.56E-01	1.223E-01	3.35E-02	3.52E-02	1.185E-04	1.134E-03	4
6 6.08E-01	1.223E-01	2.63E-02	2.85E-02	-1.135E-05	1.159E-03	6
8 1.09E-00	1.223E-01	2.17E-02	2.20E-02	-1.449E-05	1.170E-03	8
1 1.49E-00	2.445E-01	1.75E-02	1.02E-02	-5.028E-05	1.182E-03	1
2 2.43E-00	2.445E-01	1.28E-02	5.37E-03	-1.277E-05	1.153E-03	2
3 4.58E-00	2.445E-01	8.09E-03	3.49E-03	-1.136E-05	1.171E-03	3
4 8.13E-00	2.445E-01	5.50E-03	2.09E-03	-1.529E-06	1.191E-03	4
6 1.69E-01	3.233E-00	4.45E-04	3.88E-04	-3.281E-06	1.096E-03	6
8 3.05E-01	3.233E-00	3.60E-04	3.15E-04	-1.895E-06	9.327E-04	8
1 5.8E-01	5.59E-00	2.27E-05	1.75E-05	-3.805E-07	2.231E-04	1
2 3.06E-01	1.117E-00	1.27E-05	6.48E-06	-1.865E-08	3.681E-04	2
3 5.8E-01	1.117E-00	6.83E-02	6.2			3

INTEGRAL (SUM) UNDER HAD SPECTRUM = 6.83E-02 M/SEC  
 AVERAGE = 2.21E-02 M/SEC  
 VARIANCE = 3.35E-03 M/SEC  
 TREND = 7.84E-04 M/SEC

SUMMARY OF ERRORS FOR THIS JOB ERROR NUMBER NUMBER OF ERRORS  
 217 1

TABLE III

CROSS-SPECTRUM STATISTICS FOR 70038  
 BETWEEN CHANNELS 1015A 0 UBC SONIC TAPE 140/1/10/9/68  
 AND 2SONIC W TAPE 140/1/10/9/68  
 STATISTICS ARE BASED ON 102 BLOCKS OF 1024 SAMPLES EACH  
 THE SAMPLING FREQUENCY WAS 62.7710SAMP/SEC  
 MAKING THE BLOCK LENGTH 16.31326 SECONDS  
 TREND IS THE AVERAGE OF  
 (VALUE(A)-VALUE(B))/(BLOCK NO.(A)-BLOCK NO.(B))  
 PHASE POSITIVE MEANS SECOND CHANNEL LAGS FIRST  
 CALIBRATION FACTORS OF -8.05CE-01 AND  
 6.540E-01 HAVE BEEN APPLIED TO THE INPUT DATA

	FREQUENCY HERTZ	CO-SPECTRUM (M/SEC	STD. DEV. *M/SEC	TREND /HERTZ
1	5.31E-02	-1.33E-C1	2.49E-01	1.56E-03
2	1.19E-01	-8.27E-C2	1.26E-01	1.60E-04
3	1.81E-01	-6.52E-C2	1.15E-01	2.68E-04
4	2.43E-01	-4.78E-C2	7.76E-02	2.89E-04
5	3.32E-01	-2.84E-C2	4.19E-02	-3.55E-05
6	4.56E-01	-2.50E-C2	3.38E-02	5.88E-05
7	6.06E-01	-1.15E-C2	1.95E-02	7.14E-05
8	8.18E-01	-1.08E-C2	1.44E-02	2.80E-05
9	1.09E-00	-7.32E-C3	9.14E-03	5.33E-05
10	1.45E-00	-3.91E-C3	4.87E-03	2.61E-05
11	1.94E-00	-2.32E-C3	3.01E-03	1.38E-05
12	2.58E-00	-9.95E-C4	1.78E-03	6.44E-06
13	3.43E-00	-6.73E-C4	8.97E-04	3.43E-06
14	4.58E-00	-4.49E-C4	5.39E-04	9.28E-07
15	6.13E-00	-2.48E-C4	2.58E-04	1.01E-06
16	8.16E-00	-1.27E-C4	1.32E-04	1.00E-06
17	1.09E-01	-4.29E-C5	7.32E-05	2.46E-07
18	1.45E-01	4.56E-C6	2.27E-05	-8.67E-08
19	1.94E-01	5.76E-C6	1.17E-05	-6.49E-08
20	2.58E-01	-2.92E-C7	4.81E-06	-2.25E-08
21	3.43E-01	-7.10E-C7	4.43E-06	-6.87E-09
INTEGRAL (SUM) UNDER COSPECTRUM = -3.35E-02 M/SEC				

TABLE IV

Integral (Sum) under Quadspectrum =  $-6.21\text{E-}03$  M/sec

TABLE V

## Fourier Coefficients First 114 Components

COEF	FREQ	AMPL	PHASE	COEF	FREQ	AMPL	PHASE
1	0.0000	3.641E	0.0000	58	2.0000	1.111E	0.0000
2	0.0125	2.411E	0.0000	59	2.0125	1.111E	0.0000
3	0.0250	1.655E	0.0000	60	2.0250	1.111E	0.0000
4	0.0375	1.132E	0.0000	61	2.0375	1.111E	0.0000
5	0.0500	0.800E	0.0000	62	2.0500	1.111E	0.0000
6	0.0625	0.637E	0.0000	63	2.0625	1.111E	0.0000
7	0.0750	0.543E	0.0000	64	2.0750	1.111E	0.0000
8	0.0875	0.491E	0.0000	65	2.0875	1.111E	0.0000
9	0.1000	0.455E	0.0000	66	2.1000	1.111E	0.0000
10	0.1125	0.431E	0.0000	67	2.1125	1.111E	0.0000
11	0.1250	0.417E	0.0000	68	2.1250	1.111E	0.0000
12	0.1375	0.410E	0.0000	69	2.1375	1.111E	0.0000
13	0.1500	0.409E	0.0000	70	2.1500	1.111E	0.0000
14	0.1625	0.413E	0.0000	71	2.1625	1.111E	0.0000
15	0.1750	0.421E	0.0000	72	2.1750	1.111E	0.0000
16	0.1875	0.432E	0.0000	73	2.1875	1.111E	0.0000
17	0.2000	0.444E	0.0000	74	2.2000	1.111E	0.0000
18	0.2125	0.456E	0.0000	75	2.2125	1.111E	0.0000
19	0.2250	0.467E	0.0000	76	2.2250	1.111E	0.0000
20	0.2375	0.477E	0.0000	77	2.2375	1.111E	0.0000
21	0.2500	0.486E	0.0000	78	2.2500	1.111E	0.0000
22	0.2625	0.494E	0.0000	79	2.2625	1.111E	0.0000
23	0.2750	0.500E	0.0000	80	2.2750	1.111E	0.0000
24	0.2875	0.505E	0.0000	81	2.2875	1.111E	0.0000
25	0.3000	0.509E	0.0000	82	2.3000	1.111E	0.0000
26	0.3125	0.512E	0.0000	83	2.3125	1.111E	0.0000
27	0.3250	0.514E	0.0000	84	2.3250	1.111E	0.0000
28	0.3375	0.515E	0.0000	85	2.3375	1.111E	0.0000
29	0.3500	0.515E	0.0000	86	2.3500	1.111E	0.0000
30	0.3625	0.514E	0.0000	87	2.3625	1.111E	0.0000
31	0.3750	0.512E	0.0000	88	2.3750	1.111E	0.0000
32	0.3875	0.509E	0.0000	89	2.3875	1.111E	0.0000
33	0.4000	0.505E	0.0000	90	2.4000	1.111E	0.0000
34	0.4125	0.500E	0.0000	91	2.4125	1.111E	0.0000
35	0.4250	0.494E	0.0000	92	2.4250	1.111E	0.0000
36	0.4375	0.486E	0.0000	93	2.4375	1.111E	0.0000
37	0.4500	0.477E	0.0000	94	2.4500	1.111E	0.0000
38	0.4625	0.467E	0.0000	95	2.4625	1.111E	0.0000
39	0.4750	0.456E	0.0000	96	2.4750	1.111E	0.0000
40	0.4875	0.444E	0.0000	97	2.4875	1.111E	0.0000
41	0.5000	0.431E	0.0000	98	2.5000	1.111E	0.0000
42	0.5125	0.417E	0.0000	99	2.5125	1.111E	0.0000
43	0.5250	0.409E	0.0000	100	2.5250	1.111E	0.0000
44	0.5375	0.400E	0.0000	101	2.5375	1.111E	0.0000
45	0.5500	0.390E	0.0000	102	2.5500	1.111E	0.0000
46	0.5625	0.379E	0.0000	103	2.5625	1.111E	0.0000
47	0.5750	0.367E	0.0000	104	2.5750	1.111E	0.0000
48	0.5875	0.354E	0.0000	105	2.5875	1.111E	0.0000
49	0.6000	0.340E	0.0000	106	2.6000	1.111E	0.0000
50	0.6125	0.325E	0.0000	107	2.6125	1.111E	0.0000
51	0.6250	0.310E	0.0000	108	2.6250	1.111E	0.0000
52	0.6375	0.294E	0.0000	109	2.6375	1.111E	0.0000
53	0.6500	0.277E	0.0000	110	2.6500	1.111E	0.0000
54	0.6625	0.260E	0.0000	111	2.6625	1.111E	0.0000
55	0.6750	0.242E	0.0000	112	2.6750	1.111E	0.0000
56	0.6875	0.224E	0.0000	113	2.6875	1.111E	0.0000
57	0.7000	0.206E	0.0000	114	2.7000	1.111E	0.0000

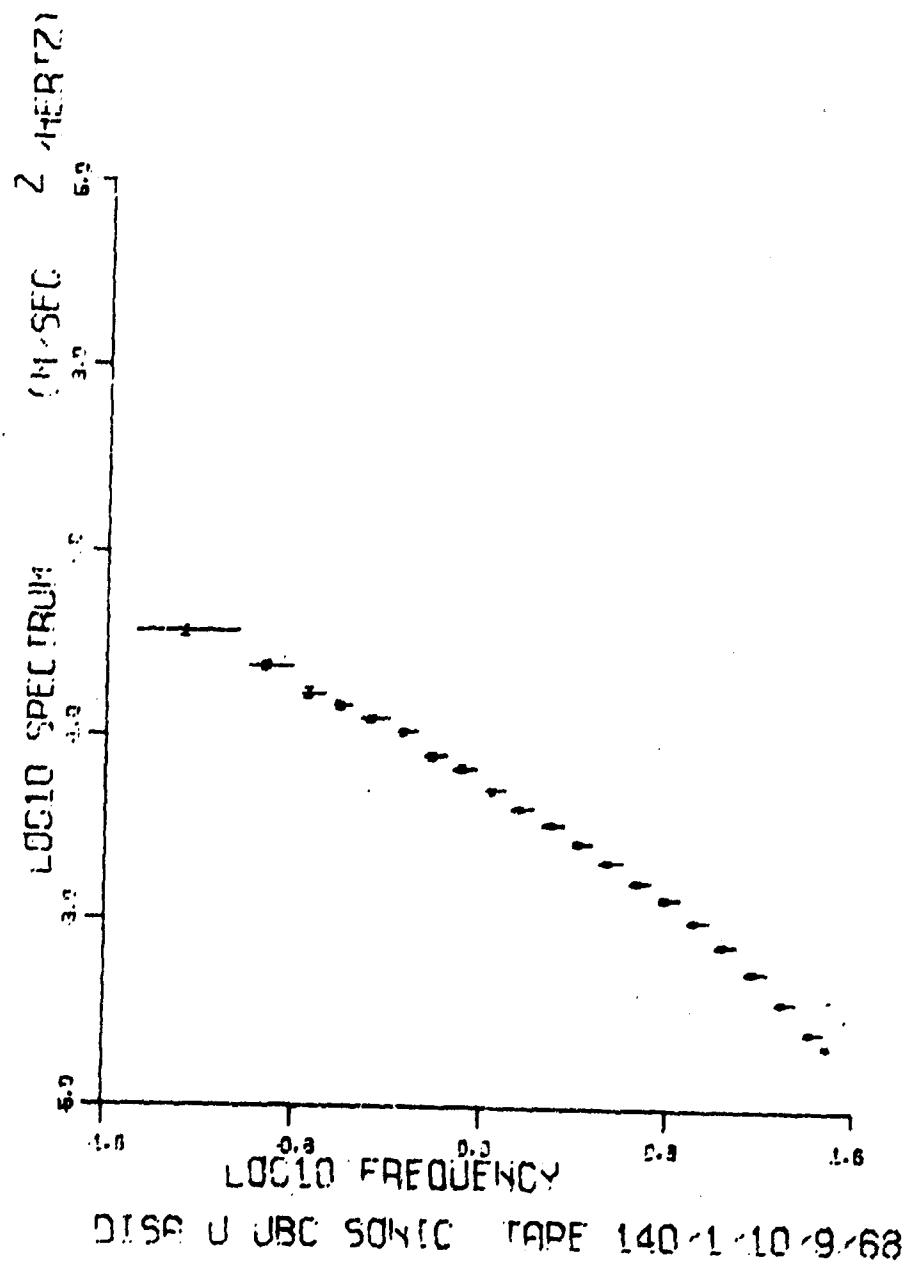


FIGURE 2. DISA U SPECTRUM

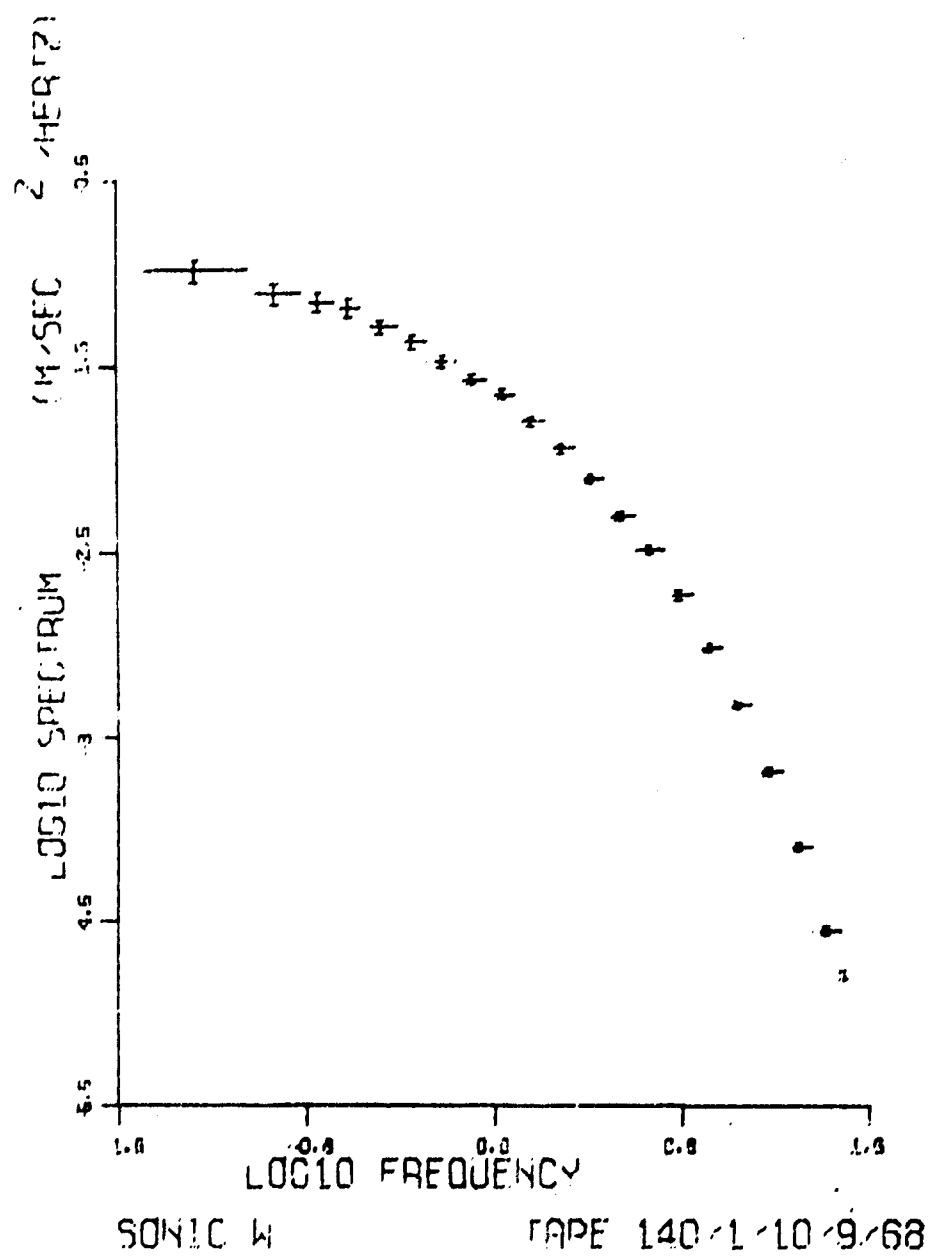


FIGURE 3. SONIC W SPECTRUM

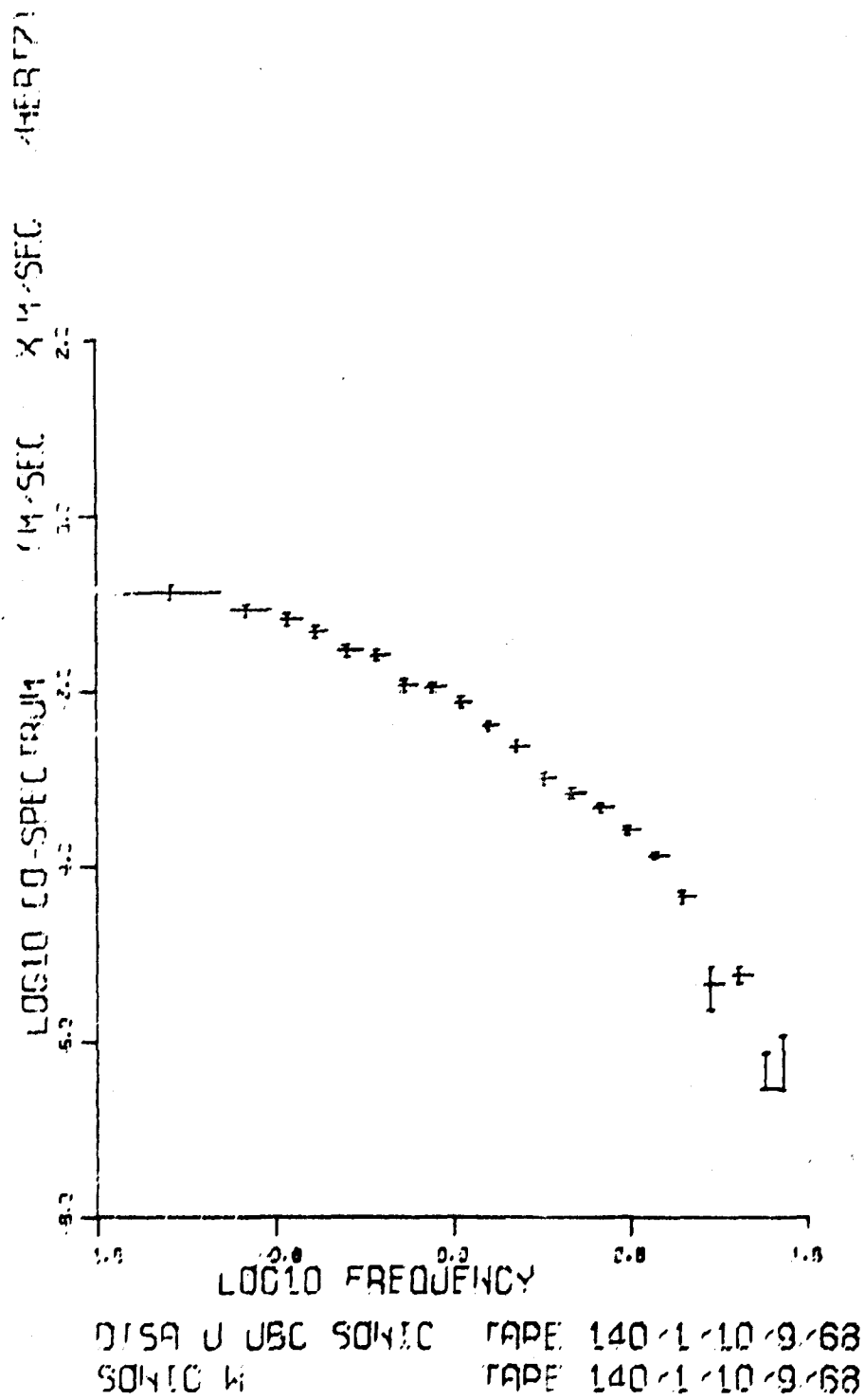
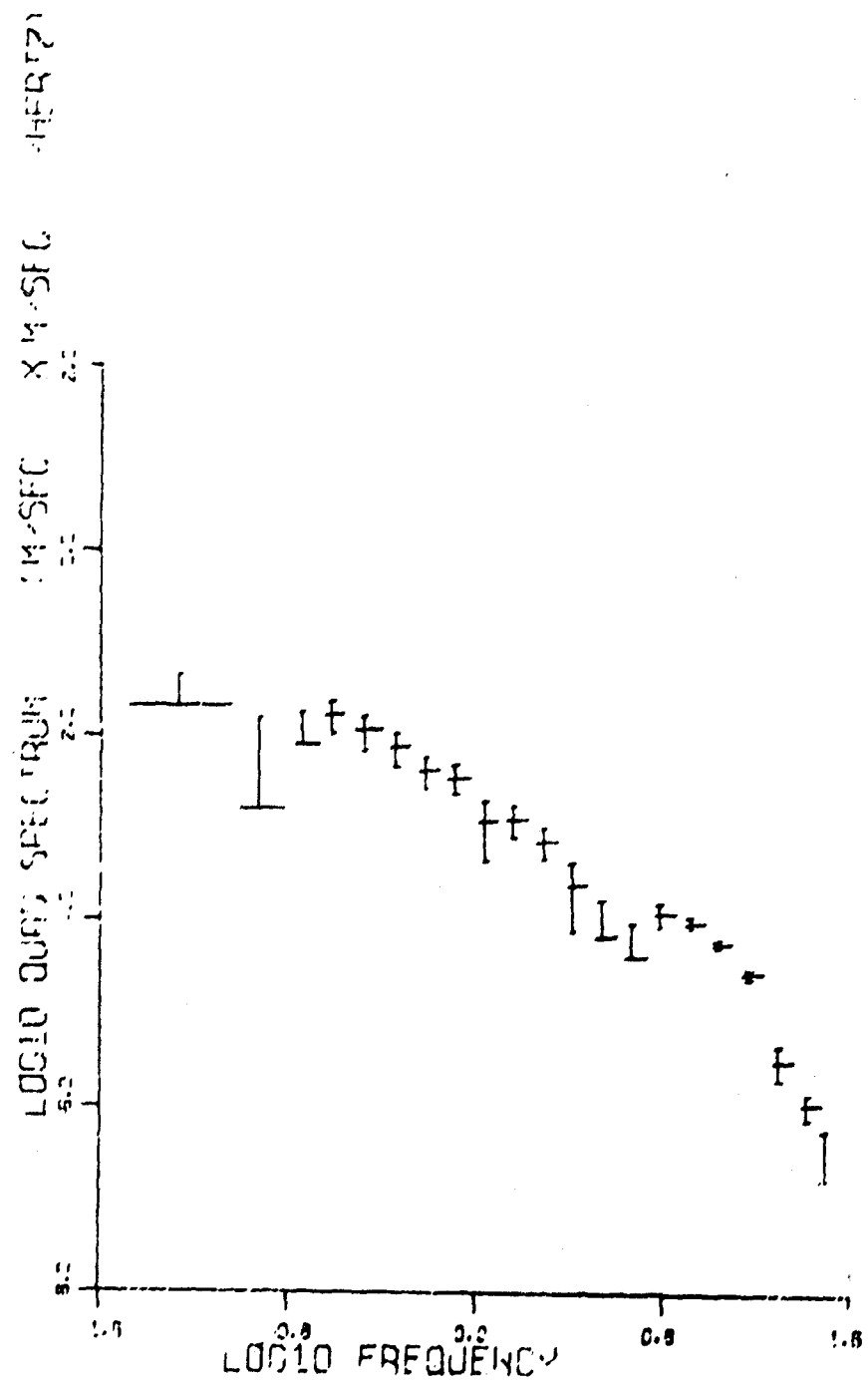


FIGURE 4. SONIC W, DISA U CO-SPECTRUM



DISA U UBC SONIC TAPE 140-1-10-9-68  
 SONIC W TAPE 140-1-10-9-68

FIGURE 5. SONIC W, DISA U QUAD-SPECTRUM

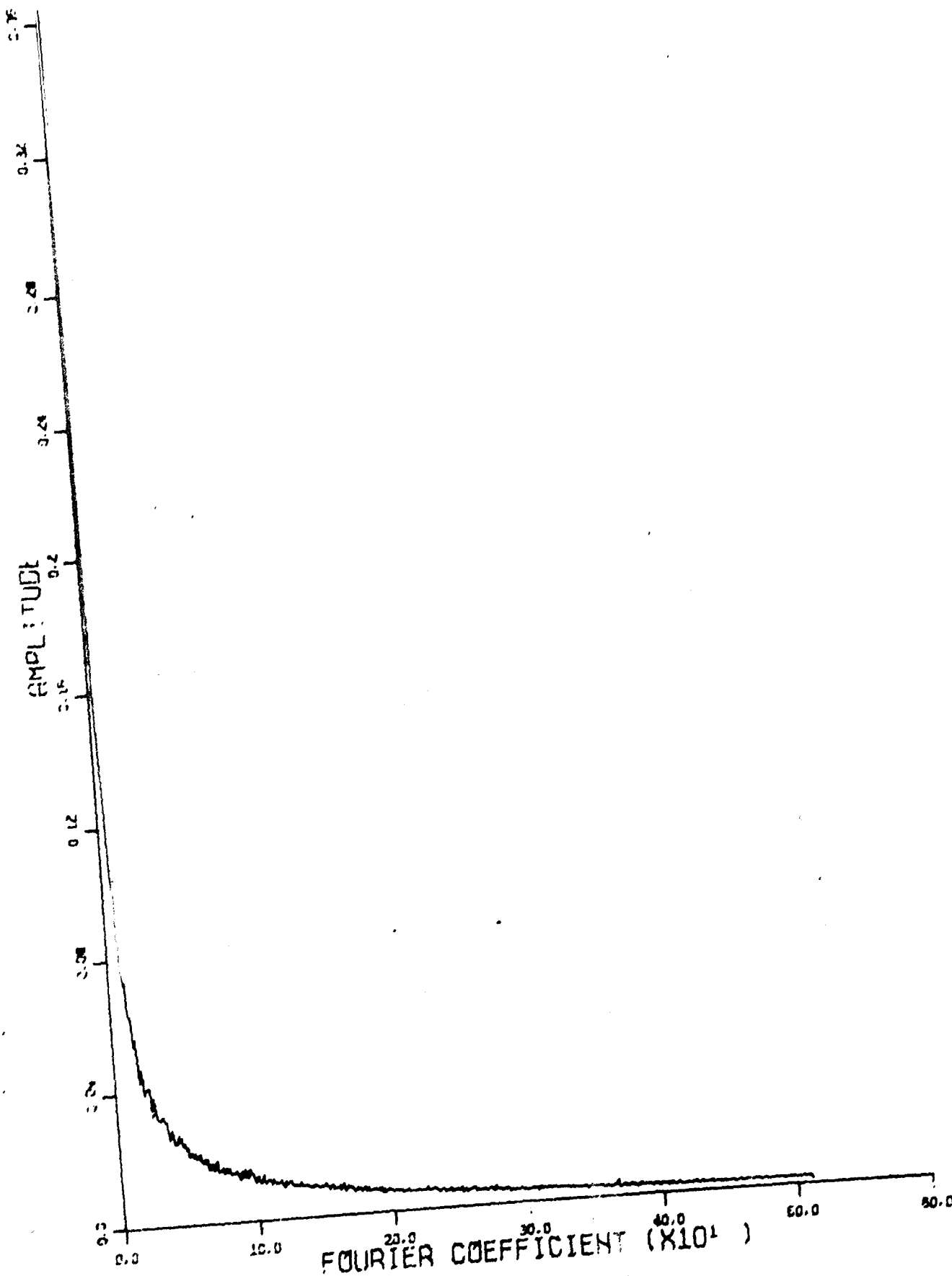


FIGURE 6. DISA U FOURIER COEFFICIENTS

#### IV INPUT DATA TAPE FORMAT

The input data tape is read at the Fortran level by an unformatted read. The read is in the subroutine OCEAN 1 and has the form:

```
READ (NUNFT) JCTR, JCHANS, (DATA(J), J = 1, JCTR)
```

where JCHANS is the number of channels digitized and JCTR is the number of data words in the tape record. DATA is an array into which the data is to be read and is single precision real and dimensioned 256.

To create a suitable tape with 4 channels of information the programmer would set JCHANS = 4 then line un-floating point data words in a single dimensioned array, DATA, in the order:

	Channel 1	DATA(1)
Cross	Channel 2	DATA(2)
Channel	Channel 3	" (3)
Sweep 1	Channel 4	" (4)
	Channel 1	" (5)
Cross	Channel 2	" (6)
Channel	Channel 3	" (7)
Sweep 2	Channel 4	" (8)
	Channel 1	" (9)
.		.
.		.
.		.
.		.
.		.
	Channel 1	.
Cross	Channel 2	.
Channel	Channel 3	.
Sweep K	Channel 4	DATA(JCTR)

where

$JCTR = 4 * K$  and is less than 256.

The statement

```
WRITE(KUNFT) JCTR, JCHANS, (DATA(J), J = 1, JCTR)
```

is then executed.

This loading of the array and writing it out is repeated until all data has been transferred to tape and then an end of file is written. It is not necessary that JCTR be the same for all records written and it may be zero. It must be however an integral number of cross channel sweeps.

## V PROCEDURE IN ANALYZING DATA

The first requirement is to produce a tape containing the data in the format that has been described. This tape is used as input to the FTOR program and a second tape is produced containing the Fourier coefficients. This tape is then used with either the SCOR or FCPLLOT program depending on the analysis desired. If co-spectra or quadrature spectra are desired then SCOR must be used. If power spectra alone are desired SCOR or FCPLLOT can be used. If more detail than SCOR supplies for the spectra, or a calcomp or printed plot of the individual Fourier coefficient amplitudes are desired, FCPLLOT should be used.

The format of the deck of control cards required to run each of the programs is described in detail on the comments cards with the Fortran source decks (APPENDIX)

The programs were originally written for an IBM 7044 (Garrett, 1967) and were then modified for an IBM 360/67 running under the Michigan Terminal System. Neither of these systems have the set of utilities for the tape handling available under OS/MUT. For this reason all tape positioning on files is handled by the programs at the Fortran level. It is therefore necessary to have a DD card present in the deck for each file on the tape whether it is read or written or just spaced over.

## VI SOURCE PROGRAMS

The system consists of three programs:

1. UBC FTOR
2. UBC SCOR - 150K in FORT step to compile.
3. UBC FCPLT

The source programs have been written as files 1, 2 and 3 in the order above on the tape NPS216. The DSNAMES are UBCFTOR, UBCSCOR and UBCFCPLT. The DCB is (RECFM = FB, LRECL = 80, BLKSIZE = 1600).

The object programs are on disc pack FAC001 in load module form. The DSNAMES for the files are F1178.TSLIB.

UBCFTOR

UBCSCOR

UBCFCPLT

#### ACKNOWLEDGEMENTS

This work was supported by the Naval Ordnance Systems Command under NAVORD Contract ORD TASK-036-005/561-1/VR104-03-01.

The time series program was set up and tested on the IBM 360/67 computer of the Naval Postgraduate School by Mr. Ron Wilson. We wish to thank the Institute of Oceanography of the University of British Columbia for making Mr. Wilson and the programs available to us.

Professor Doug Williams, Director of the Computing Facility of the Naval Postgraduate School, and his staff assisted us greatly in seeing that this work went smoothly.

#### REFERENCES

- Blackman, R. B. and J. W. Tukey, 1958. The Measurement of Power Spectra, Dover Press, New York.
- Cooley, J. W. and J. W. Tukey, 1965. "An Algorithm for the Machine Calculation of Complex Fourier Series". Mathematics of Computation, 19 (90), 297-301.
- Garrett, John. Interim Report on Digital Data Analysis. December, 1967. Internally distributed report. Institute of Oceanography, U.B.C. Canada.

## APPENDIX 1

### ANALYZING DATA DIGITIZED ON THE SDS-9300

Two computer programs are required to develop the capability to use the SDS-9300 and the associated analog computer to digitize data to be analyzed by the time series programs.

The first is an SDS-9300 program to control the digitization procedure and produce a 7-track magnetic tape containing the digital data. The second program is an IBM 360 program to convert the 7-track tape to a 9-track tape in the format described in this report for the input tape to the time series program.

The SDS-9300 program should do approximately the following:

1. It should allow for manual starting and stopping of the digitization process by the computer operator when provided with suitable signals (audio tones, oscilloscope signal changes, verbal instructions from the person concerned with the analog signal, etc.)
2. It should allow for a variable number of analog input channels.  
This number should be an input to the program at execution time.  
Ten would be a safe maximum since the time series system of programs can handle only ten.
3. The program should be written so that when the interrupt is given all channels are sampled as rapidly as possible and the data accumulated for tape output. Only after a cross channel sequence is completed should a test and possible tape output occur if enough data has been accumulated.
4. Care should be taken to make the program efficient in the sense that the computer is always ready to service the clock interrupts

when they occur. Further, care should be taken to make the records long enough that the tape is used efficiently.

5. The program for the IBM 360/67 should take this 7-track tape and produce a 9-track tape in the format required by the time series programs.
6. It is anticipated that multifile tapes will be necessary to avoid piling up large numbers of magnetic tapes with only the first part of the tape used. Therefore these two programs should be made capable of positioning themselves in multifile tapes under the control of either the job control language or input control cards to the program.

APPENDIX II  
FORTRAN SOURCE PROGRAMS

C FOURIER TRANSFORM OF TIME SERIES DATA SUPPLIED THROUGH  
 C 'OCEAN' SUBROUTINES USING P-K FORT FAST FOURIER TRANSFORM  
 C SUBROUTINE (SHARE SDA3455) IN THE IBM SYSTEM 360 MODEL 67.  
 C  
 C LAST REVISION JANUARY 24, 1969 JOHN GARRETT  
 C  
 C NBLCK BLOCKS OF 2\*\*NPOW SAMPLES EACH ARE READ FOR KCHAN OF THE  
 C NCHAN CHANNELS AVAILABLE TO THE 'OCEAN' SUBROUTINES. THE CONTENTS OF  
 C ANY CHANNEL MAY BE REPLACED BY A LINEAR COMBINATION OF ITSELF WITH ANY  
 C OTHER CHANNEL.  
 C  
 C FOR EACH BLOCK 2\*\*NPOW-1 COMPLEX FOURIER COEFFICIENTS ARE  
 C COMPUTED FOR EACH OF THE KCHAN CHANNELS. THESE ARE THEN WRITTEN ON THE  
 C OUTPUT (TAPE) 03 IN THE FORMAT DESCRIBED BELOW. IN ADDITION THE  
 C COEFFICIENTS MAY BE SUMMED IN GROUPS OF (2\*\*NPOW-1)/32 AND  
 C PRINTED OUT.  
 C  
 C ADDITIONAL OPTIONS ARE DESCRIBED UNDER THE RELEVANT CONTROL  
 C PARAMETERS BELOW.  
 C  
 C IT SHOULD BE NOTED THAT THE COEFFICIENTS PRODUCED ARE THOSE OF THE  
 C FOURIER SERIES  

$$Y(J) = \sum_{K=0}^{N/2} C(K) \exp(i 2\pi J K / N)$$
 C WITH  $J = 0, N-1, Y(J)$  REAL, AND  $I = \sqrt{-1}$   
 C  
 C THE FOLLOWING SUBROUTINES ARE REQUIRED  
 C OCEAN1, OCEAN2, OCEAN3, RWUNLD  
 C SKPFL  
 C CONVCL  
 C USCRMB  
 C P-K FORT  
 C  
 C THE FOLLOWING LOGICAL INPUT/OUTPUT DEVICES ARE USED IN THIS PROGRAM  
 C 2 = SCRATCH TAPE FOR TEMPORARY STORAGE OF COEFFICIENTS IF  
 C LOFR=1 BELOW  
 C 3 = OUTPUT (TAPE) FOR COEFFICIENTS  
 C 5 = (CARD) INPUT FOR CONTROL PARAMETERS  
 C 6 = PRINTED OUTPUT  
 C INUNIT = INPUT TAPE OF TIME SERIES DATA FOR 'OCEAN' SUBROUTINES  
 C  
 C INPUT INFORMATION REQUIRED  
 C FIRST DATA CARD IN COLUMN NUMBER  
 C 1-9 IDUSER = USER IDENTIFICATION NUMBER (9-DIGIT INTEGER)  
 C 14-15 NCHAN = NUMBER OF CHANNELS DIGITIZED ON OCEAN TAPE  
 C 25 NTYPE = (NOT RELEVANT. SET TO ZERO.)  
 C 34-35 INFILE = FILE NUMBER OF DATA ON OCEAN TAPE

C 44-45 INUNIT = NUMBER OF UNIT ON WHICH INPUT TAPE IS MOUNTED  
 C 55 NSEARH = (NOT RELEVANT. SET TO ZERO.)  
 C 61-70 SAMFRQ = SAMPLING FREQUENCY OF DIGITIZING, (SAMPLES/SECOND)  
 C (MUST INCLUDE A DECIMAL POINT)  
 C SECOND DATA CARD  
 C 4-5 NBLOCK = NUMBER OF BLOCKS DESIRED  
 C 14-15 NPOW-- MAXIMUM NUMBERS OF SAMPLES PER BLOCK WILL BE 2\*\*NPOW,  
 C MAX NPOW IS 13 (8192 SAMPLES/BLOCK) BUT NPOW WILL BE  
 C REDUCED UNTIL (2\*\*NPOW)\*KCHAN WILL FIT IN MEMORY. FOR  
 C KCHAN OF 10 THIS WILL GIVE NPOW = 10 (1024 SAMPLES/BLOCK).  
 C 24-25 MTAPE =+1 FOR OUTPUT TAPE TO BE WRITTEN  
 C =-1 FOR NO OUTPUT TAPE  
 C 34-35 NFILE - OUTPUT COEFFICIENTS WILL BE NFILE-TH FILE ON TAPE  
 C 43-45 MAXERR = (NOT RELEVANT. SET TO ZERO.)  
 C 54-55 MPRINT = -1 SUPPRESSES SUMMARY COEFFICIENTS PRINT OUT  
 C = 0 OR GREATER PERMITS PRINT OUT  
 C THIRD DATA CARD  
 C 4-5 KCHAN = NUMBER OF CHANNELS TO BE TRANSFORMED (MAX 10)  
 C 15 LOFR = 2 IF COEFFICIENTS TO BE COMPUTED FROM DATA SMOOTHED  
 C AND SUBSAMPLED (DECIMATED) USING CONVOL SUBROUTINE.  
 C WEIGHTS USED IN SMOOTHING AND DECIMATING FACTOR ARE  
 C DETERMINED BY CHOICE OF CONVOL USED.  
 C = 1 IF ALTERNATE BLOCKS TO BE MADE UP OF SAMPLES FROM DATA  
 C SMOOTHED AND DECIMATED USING CONVOL SUBROUTINE.  
 C COEFFICIENTS WILL APPEAR ON OUTPUT TAPE IN FILE  
 C IMMEDIATELY FOLLOWING THAT CONTAINING RESULTS FROM  
 C UNSMOOTHED DATA  
 C = 0 IF DATA TO BE LEFT ALONE  
 C 25 IHANN = 1 IF FOURIER COEFFICIENTS TO BE HANNED AND  
 C NORMALIZED (\*SQRT(8/3))  
 C = 0 IF NOT  
 C NEXT KCHAN CARDS  
 C 1-5 NO. OF PRIMARY A TO D CHANNEL  
 C 6-10 NO. OF SECONDARY A TO D CHANNEL  
 C 11-20 CALIBRATION ASSOCIATED WITH THE PRIMARY A TO D CHANNEL  
 C 21-30 CALIBRATION ASSOCIATED WITH THE SECONDARY A TO D CHANNEL  
 C 31-66 ALPHAMERIC NAME OF RESULTING PRIMARY CHANNEL  
 C 71-78 8 CHARACTER NAME OF UNITS FOR RESULTING PRIMARY CHANNEL  
 C THE DATA TRANSFORMED AS THE PRIMARY CHANNEL IS THEN  
 C CALIBRATION1 X VALUE OF PRIMARY + CALIBRATION2 X VALUE OF  
 C SECONDARY CHANNEL  
 C A SECOND SET OF DATA CARDS WILL PRODUCE A SECOND ANALYSIS. A BLANK  
 C CARD TERMINATES THE RUN.

FORMAT OF OUTPUT TAPE FOR EACH BLOCK TRANSFORMED IS AS FOLLOWS  
FIRST LOGICAL RECORD IS AN ARRAY OF 256 WORDS CALLED

ARTAPE(4) = NTAP (SEE BELOW)

ARTAPE(1) = IDUSER

ARTAPE

ARTAPE(3) = NUMBER OF SAMPLES /BLOCK

ARTAPE(2) = BLOCK NUMBER

ARTAPE(10) = IHANN

ARTAPE(5) = NUMBER OF CHANNELS TRANSFORMED

ARTAPE(6) = SAMPLING FREQUENCY

ARTAPE(10+K) = MCHAN(K)

ARTAPE(31+9(K-1)) = ACHNAM(K) (9A4)

ARTAPE(121+2(K-1)) = AUNITS(1,K) (2A4)

ARTAPE(140+K) = CAL(K)

NEXT NTAP LOGICAL RECORDS OF 256 WORDS EACH CALLED  
TAPRAY AND CONSIST OF

1 WORD CONTAINING INTEGER HARMONIC NO (0 TO 1BLOCK/2) FOLLOWED BY

2\*KCHAN WORDS CONTAINING FOR EACH OF THE KCHAN CHANNELS THE

REAL PART OF THE FOURIER COEFF AT THIS HARMONIC (FIRST WORD)

FOLLOWED BY THE IMAGINARY PART (SECOND WORD).

ONLY COMPLETE SEQUENCES OF 1+(2\*KCHAN) WORDS ARE INCLUDED IN A

TAPRAY SO THAT THE LAST FEW WORDS MAY CONTAIN ZEROS

AN END OF FILE IS WRITTEN ON THE OUTPUT TAPE AT THE END OF A SEQUENCE  
OF BLOCKS

J.I. GARRETT

J.P. WILSON

INSTITUTE OF OCEANOGRAPHY

UNIVERSITY OF BRITISH COLUMBIA

VANCOUVER 8, CANADA

DIMENSION W(10242), S(4100), ARTAPE(256), TAPRAY(256), MCHAN(10),  
1 CAL(10), AUNITS(2,10), ACHNAM(9,10), INDCH(12), I2TAB(14), INCR(10),  
2 INDATA(12), W1(1024), W2(1024), W3(1024), W4(1024), W5(1024),  
3 W6(1024), W7(1024), W8(1024), W9(1024), W10(1024), A(256), ATA(12)  
4 VAL(50,10)

DIMENSION MCHAN2(10), CAL2(10), DDFREQ(20)

COMMON /S, NAN, KOUNT

EQUIVALENCE (W(1), W1(1)), (W(1025), W2(1)), (W(2049), W3(1)),

1 (W(3073), W4(1)), (W(4097), W5(1)), (W(5121), W6(1)), (W(6145), W7(1)),

2 (W(7169), W8(1)), (W(8193), W9(1)), (W(9217), W10(1)), (ARTAPE(1), A(1))

3, (A(1), IDUSER), (A(2), NBLOCK), (A(3), 1BLOCK), (A(4), NTAP),

4 (A(5), KCHAN), (A(6), SAMFRQ), (A(11), MCHAN(1)),

5 (A(12), ACHNAM(1,1)), (A(121), AUNITS(1,1)), (A(141), CAL(1)),

6 (A(151), INDCH(1)), (A(163), I2TAB(1)), (A(180), INCR(1)),

7 (A(190), INDATA(1)), (A(210), TAPRAY(1)), (A(210), IAR),

8 (A(190), ATA(1))

```

      NZBLCK = 0
      CORNB = 1./SQRT(6.0)
      CURNA = 2.0 * CORNB
      IOFPCS = 0
6000 READ (5,5,END=6010) IDUSER,NCHAN,NTYPE,INFILE,INUNIT,NSEARH,SAMFRQ
      IF (INFILE.EQ.0) GO TO 6010
      CFREQ = SAMFRQ
      DO 10010 I=1,10
        MCHAN(I) = 0
        MCHAN2(I)=0
        CAL(I)=0.
        CAL2(I)=0.
      DO 10009 J=1,9
10009 ACHNAM(J,I) = 0
      DO 10010 J=1,2
        AUNITS(J,I) = 0
10010 CONTINUE
      DO 10012 I=1,12
10012 INDCH(I) = 0
      READ (5,6) NBLOCK,NPOW,MTAPE,KFILE,MAXERR,MPRINT,IRWCTL
      IF (IOFPCS.EQ.0) NFILE = KFILE
      READ(5,6) KCHAN,LOFR,IHANN
      ARTAPE(10) = IHANN
      IF (LOFR.EQ.1) REWIND 2
      DO 8 K = 1,KCHAN
        READ (5,7) MCHAN(K),MCHAN2(K),CAL(K),CAL2(K),(ACHNAM(L,K),L=1,9),
1(AUNITS(L,K),L=1,2)
        IF (MCHAN2(K).EQ.0) MCHAN2(K) = 1
        IF (MCHAN(K).EQ.0) GO TO 9
        KCH = MCHAN(K)
        INDCH(KCH) = K
      8 KCH = K
      9 KCHAN = KCH
      C SET UP TWCS TABLE
      DO 10 N =1,14
10 12TAB(N) = 2**N
      C DETERMINE BLOCK SIZE
11 LOKTES = KCHAN*12TAB(NPOW)
      IF(LOKTES.LE.10242) GO TO 15
      NPOW = NPCW-1
      GO TO 11
15 IBLOCK = 12TAB(NPOW)
      KBLOCK = IBLOCK/2
      KCH = 256/(1+(2*KCHAN))
      NTAP = (KBLOCK + KCH -1)/KCH
      C COMPUTE SINE TABLE
      CALL FORT(W,NPOW,S,0,IFERR)
      C POSITION OUTPUT TAPE

```

```

      IF (MTAPE.LT.0) GO TO 31
      IF (IOFPOS.NE.0) GO TO 31
      REWIND 3
      IOFPCS = 1
      IF (KFILE.GT.1) CALL SKPFL (KFILE-1,3)
C SET UP INPUT TAPE
  31 CALL OCEAN1 (NCHAN,INUNIT,MAXERR,INFILE,NTYPE,NSEARH)
      WRITE(6,35) IDUSER,NCHAN,KCHAN,NBLOCK,IBLOCK,SAMFRQ
      IF (IHANN.GT.0) WRITE(6,36)
      IF (LOFR.GT.0) CALL CONVOL (VAL,MXWAY,LDEC,1,SUM)
      DECIM = LDEC
      LINC = IBLOCK
      IF (IBLOCK.LT.1024) LINC = 1024
C SET UP STORAGE REFERENCES
      DO 40 K = 1,KCHAN
      KCH = K - 1
  40  INCR(K) = LINC * KCH
      MINCR = LINC
      LOBLO = LOFR
      MBLOCK = 1
C READ SOME DATA
  12  NPAR = 0
      IADD = 0
      N111 = 0
      7000 NOLD = 1
          SAMFRQ = DFREQ
          IF (LOBLO.GT.0) SAMFRQ = DFREQ/DECIM
          DO 289 IB = 1,IBLOCK
  48  IF (LOBLO) 49,49,101
  101 DO 150 LSAMP=1,LDEC
          CALL OCEAN2 (NIND,KBLOCK,INDATA(1))
          NIND = NIND+1
          GO TO (140,70,116,105,140),NIND
  105 N111 = N111+1
          KCH = INDATA(1)/NCHAN
          ICH = INDATA(1)-(KCH*NCHAN)
          IF (ICH) 51,106,51
  106 DO 108 IC = 1,KCH
          DO 107 K = 1,KCHAN
  107 VAL(NOLD,K) = 512.
          NOLD = NOLD+1
  108 CONTINUE
          GO TO 150
  116 IF (NZBLOK.EQ.KBLOCK) GO TO 7000
          NPAR = NPAR + 1
          NZBLOK = KBLOCK
          IF (NPAR.GT.MAXERR) GO TO 5000
          GO TO 7000

```

```

140 DO 145 K = 1,KCHAN
    IDSUB = MCHAN(K)
    IDSUB2 = MCHAN2(K)
    IF (NIND.NE.5) GO TO 7020
    DATA = ATA(IDSUB)*CAL(K) + ATA(IDSUB2)*CAL2(K)
    GO TO 7030
7020 DATA = (FLOAT(INDATA(IDSUB))/100. - 5.11)*CAL(K)
    DATA = DATA + (FLOAT(INDATA(IDSUB2))/100. - 5.11)*CAL2(K)
7030 VAL(NOLD,K) = DATA
145 CONTINUE
    NOLD = NOLD+1
    IF(NOLD.GT.MXWAY) NOLD = 1
150 CONTINUE
    IF(IADD.EQ.0.AND.NOLD.GT.1) GO TO 101
    DO 160 K = 1,KCHAN
        IDSUB = MCHAN(K)
        CALL CONVOL(VAL,NOLD,K,2,SUM)
        ATA(IDSUB) = SUM
160 CONTINUE
    NIND = 6
    GO TO 100
49 CALL OCEAN2(NIND,KBLOCK,INDATA(1))
C NIND = 1 MEANS END OF FILE, 2 MEANS PARITY ERROR, 3 MEANS ONES SEARCH
C NIND = 4 MEANS OCEAN2 WILL RETURN A FLOATING POINT NUMBER BETWEEN
C 0. AND 1023.
    NIND = NIND + 1
    GO TO (100,70,60,50,100) , NIND
50 N111 = N111 + 1
    KCH = INDATA(1)/NCHAN
    ICH = INDATA(1) - (KCH*NCHAN)
    IF (ICH) 51,56,51
51 WRITE(6,55) MBLOCK,KBLOCK,IADD
    GO TO 5000
56 DO 58 IC = 1,KCH
    IADD = IADD + 1
    DO 58 K = 1,KCHAN
    IWSUB = INCR(K) + IADD
58 W(IWSUB) = 0.0
    GO TO 100
60 IF (NZBLOK.EQ.KBLOCK) GO TO 7000
    NPAR = NPAR + 1
    NZBLOK = KBLOCK
    IF (NPAR.GT.MAXERR) GO TO 5000
    GO TO 7000
70 WRITE(6,75) MBLOCK,KBLOCK
    GO TO 5000
C STORE DATA IN PSFUDO 2-D ARRAY
100 IADD = IADD + 1

```

```

DO 200 K = 1,KCHAN
  IDSUB = MCHAN(K)
  IDSUB2 = MCHAN2(K)
  IWSUB = INCR(K)+IADD
  IF (NIND.LT.5) GO TO 7040
  IF (NIND.EQ.5) DATA = ATA(IDSUB)*CAL(K) + ATA(IDSUB2)*CAL2(K)
  IF (NIND.EQ.6) DATA = ATA(IDSUB)
  GO TO 7050
7040 DATA = (FLOAT(INDATA(IDSUB))/100. - 5.11)*CAL(K)
  DATA = DATA + (FLOAT(INDATA(IDSUB2))/100. - 5.11)*CAL2(K)
7050 W(IWSUB) = DATA
200 CONTINUE
  IF (IADD.GE.IBLOCK) GO TO 299
289 CONTINUE
C NOW GET FOURIER TRANSFORM
299 MPOW = NPOW - 1
  CALL FORT(W1,MPOW,S,-2,IFERR)
  CALL USCRMB(W1,IBLOCK,S)
  IF (KCHAN.LE.1) GO TO 340
  KCH = MINCR/1024
  GO TO (320,300,300,301),KCH
300 KCH = KCHAN - 1
  GO TO (310,309,308,307), KCH
301 CALL FORT(W5,MPOW,S,-2,IFERR)
  CALL USCRMB(W5,IBLOCK,S)
  GO TO 340
320 KCH = KCHAN/2
  GO TO (306,305,304,303,302), KCH
302 CALL FORT(W10,MPOW,S,-2,IFERR)
  CALL USCRMB(W10,IBLOCK,S)
303 CALL FORT(W8,MPOW,S,-2,IFERR)
  CALL USCRMB(W8,IBLOCK,S)
304 CALL FORT(W6,MPOW,S,-2,IFERR)
  CALL USCRMB(W6,IBLOCK,S)
305 CALL FORT(W4,MPOW,S,-2,IFERR)
  CALL USCRMB(W4,IBLOCK,S)
306 CALL FORT(W2,MPOW,S,-2,IFERR)
  CALL USCRMB(W2,IBLOCK,S)
  KCH = (KCHAN+1)/2
  GO TO (340,310,309,308,307),KCH
330 KCH = (KCHAN-1)/2
  GO TO (310,309,308,307), KCH
307 CALL FORT(W9,MPOW,S,-2,IFERR)
  CALL USCRMB(W9,IBLOCK,S)
308 CALL FORT(W7,MPOW,S,-2,IFERR)
  CALL USCRMB(W7,IBLOCK,S)
309 CALL FORT(W5,MPOW,S,-2,IFERR)
  CALL USCRMB(W5,IBLOCK,S)

```

```

310  CALL FORT(W3,MPOW,S,-2,IFERR)
      CALL USCRM8(W3,IBLOCK,S)
340  IF (IFERR) 350,500,350
350  WRITE (6,355) IFERR
C COEFFICIENTS NOW UNSCRAMBLD,NOW BEGIN SUM AND PRINT
500  IF (IHANN.LE.0) GO TO 880
      DO 800 K = 1,KCHAN
          ISUB = INCR(K) + 1
          ISTAR = INCR(K) + 5
          ILSUB = INCR(K) + IBLOCK - 2
          KLR = ISUB + 2
          KLM = ISUB + 3
          AR3 = 0.0
          AR2 = W(ISUB)
          AR1 = W(KLR)
          AM3 = 0.0
          AM2 = 0.0
          AM1 = W(KLM)
          W(ISUB) = CORNA*(AR2-AR1)
          DO 700 KLP = ISTAR,ILSUB,2
              KM = KLP+1
              AR3 = AR2
              AR2 = AR1
              AR1 = W(KLP)
              AM3 = AM2
              AM2 = AM1
              AM1 = W(KM)
              W(KLR) = CORNB*((2.*AR2)-AR3-AR1)
              W(KLM) = CORNB*((2.*AM2)-AM3-AM1)
              KLR = KLP
              KLM = KM
700  CONTINUE
          ISUB = INCR(K)+2
          AR3 = AR2
          AR2 = AR1
          AR1 = W(ISUB)
          W(KLR) = CORNB*((2.*AR2)-AR3-AR1)
          W(ISUB) = CORNA*(AR1-AR2)
          AM3 = AM2
          AM2 = AM1
          AM1 = 0.
          W(KLM) = CORNB*((2.*AR2)-AR3-AR1)
800  CONTINUE
880  IF (MPRINT.LT.0) GO TO 1300
      KCH = NPOW-6
      IF (KCH.LE.0) KCH = 1
      ISUM = I2TAB(KCH)
      KFIRST = 1

```

```

          KLAST = 5
900      IF(KCHAN.LT.KLAST)KLAST = KCHAN
          WRITE(6,905)IDUSER,MBLOCK,IBLOCK,TODAY,NPAR,N111
          IF(IHANN.GT.0) WRITE(6,36)
          DO 920 K = KFIRST,KLAST
          WRITE(6,915) K,MCHAN(K),(ACHNAM(L,K),L= 1,9),(AUNIT(L,K),L=1,2),CAL(K)
920      CONTINUE
          WRITE(6,925) (MCHAN(K),K=KFIRST,KLAST.
          WRITE (6,935)
          SU = 1
          IHAR = -1
          DO 1200 ICD = 1,33
          DO 950 K= KFIRST,KLAST
          KL = 2*K
          TAPRAY(KL)= 0.0
950      TAPRAY(KL+1) = 0.0
          DO 990 IS = 1,ISUM
          IHAR = IHAR+1
          IADD = 2*IHAR
          DO 980 K=KFIRST,KLAST
          KL = 2*K
          LINC = INCR(K)
          ISTORE = LINC+IADD
          IF(IHAR.EQ.(IBLOCK/2)) GO TO 970
          TAPRAY(KL) = TAPRAY(KL) + W(ISTORE +1)
          IF (IHAR.EQ.0) GO TO 980
          TAPRAY(KL+1) = TAPRAY(KL+1)+W(ISTORE+2)
          GO TO 980
970      TAPRAY(KL) = TAPRAY(KL)+W(LINC+2)
980      CONTINUE
          IF (IHAR.EQ.0) GO TO 1000
990      CONTINUE
          SU = ISUM
1000     KONE = 2*KFIRST
          KTWO =(2*KLAST)+1
          DO 1010 KL = KONE,KTWO
          TAPRAY(KL) = TAPRAY(KL)/SU
1010     WRITE(6,1105) IHAR,(TAPRAY(KL),KL=KONE,KTWO)
1200     CONTINUE
          IF (KLAST.GE.KCHAN ) GO TO 1300
          KFIRST = KLAST + 1
          KLAST = KFIRST + 4
          GO TO 900
C END OF SUM AND PRINT LOOP, NOW WRITE OUT PUT TAPE
1300     IF(MTAPE)1350,1301,1301
1301     IF(LCBLO-1) 1302,1303,1302
1302     WRITE (3) ARTAPE
          GO TO 1304

```

```

1303 WRITE (3) ARTAFE
1304 CONTINUE
C SET UP 256 WORD TAPRAY
  IHARM=(IBLOCK/2)-1
  MAXTAP = 256-(2*KCHAN)
C ZERO TH HARMONIC IS A SPECIAL CASE
  IHAR = 1
  IAR = 0
  DO 1310 K = 1,KCHAN
    ISUB = INCR(K)
    IHAR = IHAR+1
    TAPRAY(IHAR) = W(ISUB+1)
    IHAR = IHAR+1
1310  TAPRAY(IHAR) = 0.0
    DO 1330 IH = 1,IHARM
      IHH = 2*IH
      IHAR = IHAR+1
      TAPRAY(IHAR) = IH
      DO 1320 K=1,KCHAN
        ISUB = IHH + INCR(K)
        TAPRAY(IHAR+1)=W(ISUB+1)
        TAPRAY(IHAR+2)=W(ISUB+2)
        IHAR = IHAR+2
1320  CONTINUE
      IF(IHAR.LT.MAXTAP)GO TO 1330
      IHAR=0
      IF (LCBLO-1) 1322,1323,1322
1322  WRITE (3) TAPRAY
      GO TO 1330
1323  WRITE (3) TAPRAY
1330  CONTINUE
C (IBLOCK/2)TH HARMONIC IS ALSO A SPECIAL CASE
  IHAR = IHAR+1
  TAPRAY(IHAR) =IHARM+1
  DO 1340 K=1,KCHAN
    ISUB = INCR(K)
    TAPRAY(IHAR+1) = W(ISUB +2)
    TAPRAY(IHAR+2) = 0.0
1340  IHAR = IHAR+2
    IF(LCBLO-1) 1341,1342,1341
1341  WRITE (3) TAPRAY
      WRITE(6,1355) MBLOCK,NFILE
      GO TO 1343
1342  WRITE (3) TAPRAY
      WRITE(6,13425) MBLOCK,IDUSER
1343  CONTINUE
1350  IF(LOBLC.NE.1) MBLOCK = MBLOCK+1
      IF(MBLOCK.GT.NBLOCK) GO TO 5000

```

```

        IF(LOFR.EQ.0) GO TO 12
        IF(LOBLO-1)1351,1352,12
1351    LOBLC = 1
        GO TO 12
1352    LOBLC = 0
        GO TO 12
C WRAP UP ROUTINES
5000    IF(MTAPE.LT.0) GO TO 5500
        END FILE 3
        IF(LOFR.NE.1) GO TO 5500
        END FILE 2
        REWIND 2
        IDTEST = IDUSER
5200    READ (3,END=5300) ARTAPE
        WRITE (3) ARTAPE
        GO TO 5200
5300    END FILE 3
        NFILE = NFILE + 1
        IDUSER = IDTEST
        WRITE(6,5305) IDUSER,NFILE
5500    CONTINUE
        IF (MTAPE.GE.0) NFILE = NFILE + 1
        WRITE (6,8001)
        WRITE (6,8002)
        CALL OCEAN3
        GO TO 6000
6010    CALL RWUNLD
        IF (MTAPE.LT.0) GO TO 6020
        IF (IRWCTL.NE.0) GO TO 6030
        REWIND 3
6020    WRITE (6,5005)
        CALL EXIT
6030    REWIND 3
        GO TO 6020
6070    WRITE(6,6075) MBLOCK,NFILE,LOBLO
6075    FORMAT(5X,3CH END OF OUTPUT TAPE, MBLOCK = ,15,9H NFILE = ,
1      15,9H LOBLC = ,15/5X,21HPROCESSING TERMINATED )
        GO TO 6010
5      FORMAT (19,1X,5(15,5X),F10.0)
6      FORMAT (7(15,5X))
7      FORMAT (215,2F10.0,9A4,4X,2A4)
35     FORMAT(46H1 FOURIER TRANSFORM OF OCEAN TIME SERIES DATA /14H USE
1R NUMBER ,111/16H OCEAN TAPE HAS,15,1X,18HCHANNELS, OF WHICH ,
2 15,16H WILL BE DONE IN ,15,10H BLOCKS OF ,16,13H SAMPLES EACH /
3 39H THE FREQUENCY OF DIGITAL SAMPLING WAS ,F10.2,9H SAMP/SFC/)
36     FORMAT(64H0 FOURIER COEFFICIENTS TO BE HANNED AND NORMALIZED (*SQR
1T(8/3))
55     FORMAT(32H CHANNEL IDENTITY LOST IN BLOCK ,15, 7H,RECORD,15,

```

```

1 6H,SWEEP,17 )
75  FORMAT(36H END OF FILE ON INPUT TAPE IN BLOCK,15,7H,RECORD,15)
355  FORMAT(15H TCTAL IFERR =, 110)
905  FORMAT(27H1 SUMMARY COEFFICIENTS FOR ,110,1H,,15,11H1H BLOCK OF,
1 17,8H SAMPLES, 10X, /5X,37HIN THIS BLOCK PARITY ERRORS ON TAPE
2= ,13,32H,ALL ONES NOT FOUND IN CHANNEL 1,14,6H TIMES //16X,
3 1HK,4X, 7HCHANNEL,17X,4HNAME,25X,5HUNITS,3X,10HCAL.FAC.0R )
915  FORMAT( 15X,13,4X,13, 5X,9A4,8X, 2A4,5X,1PE9.2)
925  FORMAT (2X,4HLAST,1X,5(8X,8HCHANNEL ,12,4X))
935  FORMAT (2X,4HHARM,3X,5(4HREAL,6X,4HIMAG,8X))
1105  FORMAT (2X,14,1X,5(3X 'PE9.2,1X,1PE'.2))
13425  FORMAT(24H COEFFICIENTS FOR BLOCK ,15,36H OF SMOOTHED AND DECIMA
1TED DATA FOR ,19,14H WRITTEN ON 02 )
1355  FORMAT(24H COEFFICIENTS FOR BLOCK ,15,17H WRITTEN IN FILE ,13,
1 15H ON OUTPUT TAPE )
5005  FORMAT(13HONORMAL EXIT )
80  FORMAT (20(1X,F5.2))
5305  FORMAT(34H0 SMOOTHED AND DECIMATED DATA FOR ,19,39H SUCCESSFULLY C
1COPIED FROM 02 ONTO FILE ,12,14H OF TAPE ON 03 /)
8001  FORMAT (42H1NORMAL COMPLETION OF A PROCESSING REQUEST)
8002  FORMAT (1H )
END

```

```

SUBROUTINE CONVOL(VAL,NOLD,KCHAN,MODE,SUM)
DIMENSION VAL(50,1),WEIGHT(50)
IF(MODE-1) 10,10,20
10  LDEC = 10
    MXWAY = 21
    WAY = 1.0/21.0
    DO 11 KM = 1,21
11  WEIGHT(KM) = WAY
    KCHAN = LDEC
    NOLD = MXWAY
    WRITE(6,5) LDEC,MXWAY,(WEIGHT(K),K=1,MXWAY)
    RETURN
20  SUM = 0.
    NCCUNT = 0
    NFIR = NOLD-1
30  NFIR = NFIR+1
    NCCUNT = NCCUNT +1
    IF(NCCUNT-MXWAY)17,17,21
17  IF(NFIR-MXWAY)40,40,18
18  NFIR = 1
40  SUM = SUM+VAL(NFIR,KCHAN)
    GO TO 30

```

```

21 SUM = SUM*WEIGHT(1)
   RETURN
5  FORMAT(50H1 ALTERNATE BLOCKS OF DATA DECIMATED BY FACTOR OF 13/
1  2X, 22H AFTER SMOOTHING WITH ,15,35H WEIGHTS WITH THE FOLLOWING VA
2LUES: /(10X,1PE14.5))
   END

```

```

SUBROUTINE USCRMB (C,M,S)
DIMENSION C(1),S(1)
N = M
ST = C(1)
C(1) = 0.5*(C(1) + C(2))
C(2) = 0.5*(ST - C(2))
K = N/2 - 1
MSIN = N/4
C(K+3) = -C(K+3)
DO 10 I=3,K,2
  IS = (I-1)/2
  IC = MSIN - IS
  ST = S(IS)
  CT = S(IC)
  A1 = C(I)
  B1 = C(I+1)
  L = N - I
  A2 = C(L+2)
  B2 = C(L+3)
  C(I) = 0.5*(A1+A2+(B1+B2)*CT-(A1-A2)*ST)
  C(I+1) = 0.5*(B1-B2-(B1+B2)*ST-(A1-A2)*CT)
  C(L+2) = 0.5*(A1+A2-(B1+B2)*CT+(A1-A2)*ST)
10 C(L+3) = 0.5*(B2-B1-(B1+B2)*ST-(A1-A2)*CT)
  RETURN
END

```

# FORTRAN OCEAN PACKAGE

THE FORTRAN OCEAN PACKAGE IS DESIGNED TO DUPLICATE THE 'OCEAN' AND 'OCEANB' SUBROUTINE PACKAGES WITH A TAPE WHICH WAS EITHER CREATED BY THE DIGITAL FILTERING SYSTEM OR IS IN THE SAME FORMAT. THE CALLS ARE EQUIVALENT.

THE CALLING SEQUENCES ARE:

CALL OCEAN1 (KCHAN,NUNIT,MAXERR,NFILE,NTYPE,NSEARH)

WHERE,

NCHAN = NO. OF CHANNELS DIGITIZED ON THE FORTRAN OCEAN TAPE  
 NUNIT = NO. OF UNIT ON WHICH THE FORTRAN OCEAN TAPE IS MOUNTED  
 MAXERR (THIS PARAMETER IS NOT APPLICABLE TO FORTRAN OCEAN BUT MUST BE INCLUDED FOR COMPATIBILITY)  
 NFILE = NO. OF DESIRED FILE ON THE FORTRAN OCEAN TAPE  
 NTYPE (NOT APPLICABLE BUT IS INCLUDED FOR COMPATIBILITY)  
 NSEARH (NOT APPLICABLE BUT IS INCLUDED FOR COMPATIBILITY)  
 CALL CCEAN2 (IND,KBLOCK,N)  
 WHERE,  
     IND = 0 FOR NORMAL RETURN (RETURNED)  
         = 1 FOR END OF INPUT FILE (RETURNED)  
         = 4 IF THE VECTOR RETURNED IS IN FLOATING POINT (FORTRAN VERSION OF OCEAN ONLY) (RETURNED)  
 KBLOCK = THE SEQUENTIAL NUMBER OF THE PHYSICAL TAPE BLOCK OF THE NEXT DATA WHICH WILL BE RETURNED BY THE NEXT OCEAN2 CALL. (RETURNED)  
 N = FIRST LOCATION OF THE ARRAY INTO WHICH THE CROSS CHANNEL SEQUENCE IS TO BE RETURNED  
 CALL OCEAN3  
 THIS CALL RESULTS IN A 1 LINE SUMMARY BEING LISTED ON LOGICAL UNIT 6. THIS MAY BE CALLED AT ANY TIME.

```

SUBROUTINE OCEAN1 (NCHAN,NUNIT,MAXERR,NFILE,NTYPE,NSEARH)
DATA KFILE/0/
KUNIT = NUNIT
IF (KFILE.GT.0) GO TO 20
10 REWIND KUNIT
KFILE = 1
KSTART = 1
20 IF (KFILE.GT.NFILE) GO TO 10
IF (KFILE.EQ.NFILE.AND.KSTART.EQ.0) GO TO 10
KSKIP = NFILE - KFILE
IF (KSKIP.LE.0) GO TO 50
DO 40 I=1,KSKIP
30 READ (NUNIT,END=40) IDUMMY
GO TO 30
40 CONTINUE
50 KFILE = NFILE
KSTART = 1
KMAX = 256
KCTR = KMAX + 1
KREAD = 0
RETURN
ENTRY OCEAN2 (IND,KBLOCK,X)
DIMENSION X(1),DATA(256)
IF (KCTR.GT.KMAX) GO TO 70
55 DO 60 I=1,NCHAN
X(I) = DATA(KCTR)
60 KCTR = KCTR + 1
IND = 4
KBLOCK = KREAD
RETURN
70 READ (KUNIT,END=80) KMAX,NCHAN,(DATA(J),J=1,KMAX)
KSTART = 0
IF (KMAX.EQ.0) GO TO 70
KCTR = 1
KREAD = KREAD + 1
GO TO 55
80 KFILE = KFILE + 1
KSTART = 1
IND = 1
RETURN
ENTRY OCEAN3
WRITE (6,4001) KREAD
RETURN
4001 FORMAT ('OCEAN 3 CALLED (FORTRAN VERSION).',I10,' TAPE BLOCKS PRO
1CESSED')
ENTRY RWUNLD
REWIND KUNIT
KFILE = 1

```

```

KSTART = 1
RETURN
END

```

THIS SUBROUTINE WHEN CALLED MOVES THE TAPE ON LOGICAL UNIT NUNIT PAST KSKIP END OF FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN FORTRAN BINARY. IF KSKIP IS ZERO OR NEGATIVE THE ROUTINE RETURNS WITHOUT MOVING THE TAPE.

```

SUBROUTINE SKPFL (KSKIP,NUNIT)
IF (KSKIP.LT.1) RETURN
DO 20 I=1,KSKIP
10 READ (NUNIT,END=20) IDUMMY
GO TO 10
20 CONTINUE
RETURN
END

```

FORT, ONE-DIMENSIONAL FINITE COMPLEX FOURIER TRANSFORM.

FORT 002  
FORT 003

```

SUBROUTINE FORT(A,M,S,IFS,IFERR)

```

FOURIER TRANSFORM SUBROUTINE, PROGRAMMED IN SYSTEM/360,  
BASIC PROGRAMMING SUPPORT, FORTRAN IV. FORM C28-6504  
THIS DECK SET UP FOR IBSYS ON IBM 7094.

FORT 004  
FORT 005  
FORT 006  
FORT 007  
FORT 008

THIS DECK MODIFIED TO ALLOW COMPUTATION OF SINE TABLE ( S(J) )  
WITH M = 14, FOR USE WITH SERIES OF 2\*\*14 REAL VALUES  
BY ADDITION OF STATEMENTS 6 AND 7 AND CHANGING 3 FROM  
IF(M-13) 5,5,2 TO IF(M-13) 5,5,6

DOES EITHER FOURIER SYNTHESIS, I.E., COMPUTES COMPLEX FOURIER SERIES  
GIVEN A VECTOR OF N COMPLEX FOURIER AMPLITUDES, OR, GIVEN A VECTOR  
OF COMPLEX DATA X DOES FOURIER ANALYSIS, COMPUTING AMPLITUDES.  
A IS A COMPLEX VECTOR OF LENGTH N=2\*\*M COMPLEX NOS. OR 2\*N REAL  
NUMBERS. A IS TO BE SET BY USER.  
M IS AN INTEGER 0.LT.M.LE.13, SET BY USER.  
S IS A VECTOR S(J)= SIN(2\*PI\*J/NP), J=1,2,...,NP/4-1,  
COMPUTED BY PROGRAM.  
IFS IS A PARAMETER TO BE SET BY USER AS FOLLOWS-  
IFS=0 TO SET NP=2\*\*M AND SET UP SINE TABLE.

FORT 009  
FORT 010  
FORT 011  
FORT 012  
FORT 013  
FORT 014  
FORT 015  
FORT 016  
FORT 017  
FORT 018  
FORT 019

IFS=1 TO SET N=NP=2\*\*M, SET UP SIN TABLE, AND DO FOURIER SYNTHESIS, REPLACING THE VECTOR A BY

$X(J) = \text{SUM OVER } K=0, N-1 \text{ OF } A(K) * \exp(2 * \pi * I / N) ** (J * K),$   
 $J=0, N-1, \text{ WHERE } I = \text{SQRT}(-1)$

THE X'S ARE STORED WITH RE X(J) IN CELL 2\*J+1  
 AND IM X(J) IN CELL 2\*J+2 FOR J=0,1,2,...,N-1.  
 THE A'S ARE STORED IN THE SAME MANNER.

IFS=-1 TO SET N=NP=2\*\*M, SET UP SIN TABLE, AND DO FOURIER ANALYSIS, TAKING THE INPUT VECTOR A AS X AND REPLACING IT BY THE A SATISFYING THE ABOVE FOURIER SERIES.  
 IFS=+2 TO DC FOURIER SYNTHESIS ONLY, WITH A PRE-COMPUTED S.  
 IFS=-2 TO DC FOURIER ANALYSIS ONLY, WITH A PRE-COMPUTED S.  
 IFERR IS SET BY PROGRAM TO-  
 =0 IF NO ERROR DETECTED.  
 =1 IF M IS OUT OF RANGE., OR, WHEN IFS=+2,-2, THE PRE-COMPUTED S TABLE IS NOT LARGE ENOUGH.  
 =-1 WHEN IFS =+1, -1, MEANS ONE IS RECOMPUTING S TABLE UNNECESSARILY.

NOTE- AS STATED ABOVE, THE MAXIMUM VALUE OF M FOR THIS PROGRAM ON THE IBM 7094 IS 13. FOR 360 MACHINES HAVING GREATER STORAGE CAPACITY, ONE MAY INCREASE THIS LIMIT BY REPLACING 13 IN STATEMENT 3 BELOW BY LOG2 N, WHERE N IS THE MAX. NO. OF COMPLEX NUMBERS ONE CAN STORE IN HIGH-SPEED CORE. ONE MUST ALSO ADD MORE DO STATEMENTS TO THE BINARY SORT ROUTINE FOLLOWING STATEMENT 24 AND CHANGE THE EQUIVALENCE STATEMENTS FOR THE K'S.

DIMENSION A(1), S(1), K(14)  
 EQUIVALENCE (K(13),K1), (K(12),K2), (K(11),K3), (K(10),K4)  
 EQUIVALENCE (K( 9),K5), (K( 8),K6), (K(7),K7), (K( 6),K8)  
 EQUIVALENCE (K( 5),K9), (K( 4),K10), (K( 3),K11), (K( 2),K12)  
 EQUIVALENCE (K( 1),K13), ( K(1),N2)

IF(M) 2,2,3

3 IF(M-13) 5,5,6

THE FOLLOWING TWO STATEMENTS ADDED BY J GARRETT , SEE ABOVE

6 IF(M-14) 7,7,2

7 IF(IFS) 2,5,2

2 IFERR=1

1 RETURN

5 IFERR=0

N=2\*\*M

IF( IABS(IFS) - 1 ) 200,200,10

WE ARE DOING TRANSFORM ONLY. SEE IF PRE-COMPUTED S TABLE IS SUFFICIENTLY LARGE

FORT 020  
 FORT 021  
 FORT 022  
 FORT 023  
 FORT 024  
 FORT 025  
 FORT 026  
 FORT 027  
 FORT 028  
 FORT 029  
 FORT 030  
 FORT 031  
 FORT 032  
 FORT 033  
 FORT 034  
 FORT 035  
 FORT 036  
 FORT 037  
 FORT 038  
 FORT 039  
 FORT 040  
 FORT 041  
 FORT 042  
 FORT 043  
 FORT 044  
 FORT 045  
 FORT 046  
 FORT 047  
 FORT 048  
 FORT 049  
 FORT 050  
 FORT 051  
 FORT 052  
 FORT 053  
 FORT 054  
 FORT 055  
 FORT 056

FORT 058  
 FORT 059  
 FORT 060  
 FORT 061  
 FORT 062  
 FORT 063  
 FORT 064

10	IF( N-NP )20,20,12	FORT	065
12	IFERR=1	FORT	066
	GO TO 200	FORT	067
C	SCRAMBLE A, BY SANDE'S METHOD	FORT	068
20	K(1)=2*N	FORT	069
	DO 22 L=2,M	FORT	070
22	K(L)=K(L-1)/2	FORT	071
	DO 24 L=M,12	FORT	072
24	K(L+1)=2	FORT	073
C	NOTE EQUIVALENCE OF KL AND K(14-L)	FORT	074
C	BINARY SORT-	FORT	075
	IJ=2	FORT	076
	DO 30 J1=2,K1,2	FORT	077
	DO 30 J2=J1,K2,K1	FORT	078
	DO 30 J3=J2,K3,K2	FORT	079
	DO 30 J4=J3,K4,K3	FORT	080
	DO 30 J5=J4,K5,K4	FORT	081
	DO 30 J6=J5,K6,K5	FORT	082
	DO 30 J7=J6,K7,K6	FORT	083
	DO 30 J8=J7,K8,K7	FORT	084
	DO 30 J9=J8,K9,K8	FORT	085
	DO 30 J10=J9,K10,K9	FORT	086
	DO 30 J11=J10,K11,K10	FORT	087
	DO 30 J12=J11,K12,K11	FORT	088
	DO 30 J1=J12,K13,K12	FORT	089
	IF(IJ-J1)28,30,30	FORT	090
28	T=A(IJ-1)	FORT	091
	A(IJ-1)=A(JI-1)	FORT	092
	A(JI-1)=T	FORT	093
	T=A(IJ)	FORT	094
	A(IJ)=A(JI)	FORT	095
	A(JI)=T	FORT	096
30	IJ=IJ+2	FORT	097
	IF(IFS)32,2,36	FORT	098
C	DOING FOURIER ANALYSIS,50 DIV. BY N AND CONJUGATE.	FORT	099
32	FN = N	FORT	100
	DO 34 I=1,N	FORT	101
	A(2*I-1) = A(2*I-1)/FN	FORT	102
34	A(2*I)=-A(2*I)/FN	FORT	103
C	SPECIAL CASE- L=1	FORT	104
36	DO 40 I=1,N,2	FORT	105
	T = A(2*I-1)	FORT	106
	A(2*I-1) = T + A(2*I+1)	FORT	107
	A(2*I+1)=T-A(2*I+1)	FORT	108
	T=A(2*I)	FORT	109
	A(2*I) = T + A(2*I+2)	FORT	110
40	A(2*I+2)= T - A(2*I+2)	FORT	111
	IF(M-1) 2,1 ,50	FORT	112

```

C SET FOR L=2
50 LEXP1=2
   LEXP1=2**(L-1)
   LEXP=8
   LEXP=2**(L+1)
   NPL= 2**MT
   NPL = NP* 2**-L
60 DO 130 L=2,M
   SPECIAL CASE- J=0
   DO 80 I=2,N2,LEXP
     I1=I + LEXP1
     I2=I1+ LEXP1
     I3 =I2+LEXP1
     T=A(I-1)
     A(I-1) = T +A(I2-1)
     A(I2-1) = T-A(I2-1)
     T =A(I)
     A(I) = T+A(I2)
     A(I2) = T-A(I2)
     T = -A(I3)
     TI = A(I3-1)
     A(I3-1) = A(I1-1) - T
     A(I3) = A(I1) - TI
     A(I1-1) = A(I1-1) +T
80 A(I1) = A(I1) +TI
   IF(L-2) 120,120,90
90 KLAST=N2-LEXP
   JJ=NPL
   DO 110 J=4,LEXP1,2
     NPJJ=NT-JJ
     UR=S(NPJJ)
     UI=S(JJ)
     ILAST=J+KLAST
     DO 100 I= J,ILAST,LEXP
       I1=I+LEXP1
       I2=I1+LEXP1
       I3=I2+LEXP1
       T=A(I2-1)*UR-A(I2)*UI
       TI=A(I2-1)*UI+A(I2)*UR
       A(I2-1)=A(I-1)-T
       A(I2) =A(I) - TI
       A(I-1) =A(I-1)+T
       A(I) =A(I)+TI
       T=-A(I3-1)*UI-A(I3)*UR
       TI=A(I3-1)*UR-A(I3)*UI
       A(I3-1)=A(I1-1)-T
       A(I3) =A(I1) -TI
       A(I1-1)=A(I1-1)+T

```

```

FORT 113
FORT 114
FORT 115
FORT 116
FORT 117
FORT 118
FORT 119
FORT 120
FORT 121
FORT 122
FORT 123
FORT 124
FORT 125
FORT 126
FORT 127
FORT 128
FORT 129
FORT 130
FORT 131
FORT 132
FORT 133
FORT 134
FORT 135
FORT 136
FORT 137
FORT 138
FORT 139
FORT 140
FORT 141
FORT 142
FORT 143
FORT 144
FORT 145
FORT 146
FORT 147
FORT 148
FORT 149
FORT 150
FORT 151
FORT 152
FORT 153
FORT 154
FORT 155
FORT 156
FORT 157
FORT 158
FORT 159
FORT 160

```

100	A(I1) =A(I1) +TI	FORT 161
	END OF I LOCP	FORT 162
110	JJ=JJ+NPL	FORT 163
	END OF J LOCP	FORT 164
120	LEXP1=2*LEXP1	FORT 165
	LEXP = 2*LEXP	FORT 166
130	NPL=NPL/2	FORT 167
	END OF L LOCP	FORT 168
140	IF(IFS)145,2,1	FORT 169
	DOING FOURIER ANALYSIS. REPLACE A BY CONJUGATE.	FORT 170
145	DO 150 I=1,N	FORT 171
150	A(2*I) =-A(2*I)	FORT 172
160	GO TO 1	FORT 173
	RETURN	FORT 174
	MAKE TABLE OF S(J)=SIN(2*PI*J/NT),J=1,2,...,NT-1,NT=NP/4	FORT 175
200	NP=N	FORT 176
	MP=M	FORT 177
	NT=N/4	FORT 178
	MT=M-2	FORT 179
	IF(MT) 260,260,205	FORT 180
205	THETA=.7853981634	FORT 181
	THETA=PI/2**((L+1) FOR L=1	FORT 182
210	JSTEP = NT	FORT 183
	JSTEP = 2**(( MT-L+1 ) FOR L=1	FORT 184
	JDIF = NT/2	FORT 185
	JDIF = 2**((MT-L) FOR L=1	FORT 186
	S(JDIF) = SIN(THETA)	FORT 187
	IF (MT-2)260,220,220	FORT 188
220	DO 250 L=2,MT	FORT 189
	THETA = THETA/2.	FORT 190
	JSTEP2 = JSTEP	FORT 191
	JSTEP = JDIF	FORT 192
	JDIF = JDIF/2	FORT 193
	S(JDIF)=SIN(THETA)	FORT 194
	JC1=NT-JDIF	FORT 195
	S(JC1)=COS(THETA)	FORT 196
	JLAST=NT-JSTEP2	FORT 197
	IF(JLAST-JSTEP)250,230,230	FORT 198
230	DO 240 J=JSTEP,JLAST,JSTEP	FORT 199
	JC=NT-J	FORT 200
	JD=J+JDIF	FORT 201
240	S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)	FORT 202
250	CONTINUE	FORT 203
260	IF(IFS)20,1,20	FORT 204
	END	FORT 205

SPECTRUM AND CROSS SPECTRUM STATISTICS FROM FOURIER  
 COEFFICIENT TAPE PRODUCED BY 'FTCR' PROGRAM ON IBM  
 SYSTEM 360 MODEL 67  
 LAST REVISION JANUARY 27, 1969 JCHN GARRETT  
 THIS PROGRAM READS FOURIER COEFFICIENTS FROM TAPE PRODUCED BY FTOR  
 PROGRAM, AND FROM THE APPROPRIATE SUMS OF THESE PRODUCE SPECTRA AND  
 COSPECTRA. A TAPE USED MAY BE FIXED WITH FREQUENCY OR MAY GO IN HALF  
 OCTAVES FROM A SPECIFIED LOW FREQUENCY. AN AVERAGE, STANDARD DEVIATION  
 AND LINEAR COEFFICIENT OF REGRESSION OVER THE ICMAX BLOCKS USED (SEE  
 FITZ DESCRIPTOR) ARE GIVEN FOR EACH VALUE OF SPECTRAL DENSITY. (SEE  
 AT EACH FREQUENCY, THE COSPECTRUM BETWEEN (1) AND (2) IS GIVEN BY  

$$(R(1)*R(2) + I(1)*I(2)) / 2.0$$
  
 AND THE CLACRATURE SPECTRUM BY  

$$(R(2)*I(1) - R(1)*I(2)) / 2.0$$
  
 WHERE (R(N) + I(N)\*SQRT(-1)) IS THE COMPLEX FOURIER COEFFICIENT  
 OF SERIES (N) AT THAT FREQUENCY  
 A VARIETY OF PLOTTED OUTPUT IS AVAILABLE. IN ALL A HORIZONTAL BAR  
 INDICATES THE FREQUENCY INTERVAL INCLUDED IN THE ESTIMATE PLOTTED, AND  
 A VERTICAL BAR INDICATES THE EXPECTED STANDARD DEVIATION OF THE  
 ESTIMATE (= STD.DEV. OF BLOCKS AVERAGED TO GIVE ESTIMATE / SQRT(NUMBER  
 OF BLOCKS))  
 THE FOLLOWING SUBROUTINES MUST BE SUPPLIED BY USER  
 PHASES  
 PLVAL, TIC, LABEL  
 THE FOLLOWING LOGICAL INPUT/OUTPUT UNITS ARE USED BY THIS PROGRAM  
 3= (TAPE) SUPPLYING COEFFICIENTS AND IDENTIFICATION AS  
 5= (CARDS) CONTROL PARAMETERS  
 6= PRINTED OUTPUT  
 THE FOLLOWING INPUT IS REQUIRED  
 A CARD IS REQUIRED TO IDENTIFY YOUR GRAPHICAL OUTPUT FOR THE COMPUTING  
 CENTRE STAFF. IT MUST BE PRESENT WHETHER PLOTS ARE PRODUCED OR NOT.  
 THE FIRST 72 COLUMNS OF THIS CARD WILL BE REPRODUCED ON THE BEGINNING  
 OF YOUR PLOT. THIS CARD APPEARS ONLY ONCE IN THE JOB AND IS THE FIRST  
 DATA CARD. THE FOLLOWING SET OF CARDS IS PRESENT FOR EACH FILE OF  
 FOURIER COEFFICIENTS TO BE PROCESSED.  
 FIRST CARD, IN COLUMN  
 1-9 USER IDENTIFICATION NUMBER FOR DATA DESIRED  
 14-15 ICMAX = NUMBER OF CHANNELS TO BE USED (MAX 10)

```

24-25 IBMX = NUMBER OF DATA BLOCKS FOR STATISTICS
34-35 I9STAR = STATISTICS START WITH BLOCK NO. I9STAR
44-45 NFILE = 0 (BLANKS) IF PLOT AXES TO BE SET BY PROGRAM
54 IAXIS = 1 IF AXES TO BE SET BY USER
55 IPLOT = 1 IF SPECTRAL DENSITY TO BE PLOTTED AGAINST FREQUENCY
        = 2 IF LOG10 SPECTRAL DENSITY TO BE PLOTTED AGAINST LOG10
        = 3 IF FREQUENCY
        = 4 IF (FREQ*SPECTRAL DENSITY) TO BE PLOTTED
        = 5 AGAINST LOG10 FREQUENCY COHERENCE AND PHASE PLOTTED
        = 6 INSTEAD OF CO AND QUAD SPECTRA
        = 7 SAME AS 6 EXCEPT COHERENCE AND PHASE PLOTTED INSTEAD
        = 8 SAME AS 3 EXCEPT COHERENCE AND PHASE INSTEAD OF CO+QU
        = 9 SAME AS 4 WITH COHERENCE AND PHASE CORRECTIONS
65 IPHASEC = 0 FOR PHASES TO BE CORRECTED.
        = 1 FOR PHASES TO BE CORRECTED.

CARL INSERTED ONLY IF IAXIS = 1
IN COLUMN
1-10 VALUE CF ORIGIN FOR SPECTRUM AND COSPECTRUM AXIS (F10.)
11-20 UNITS PER INCH FOR SPECTRUM AND COSPECTRUM AXIS (F10.)
21-30 VALUE CF ORIGIN FOR QUAD SPECTRUM AXIS (F10.)
31-40 UNITS PER INCH FOR QUAD SPECTRUM AXIS (F10.)
41-50 VALUE CF ORIGIN FOR LONG. IF PLOTTED INSTEAD OF CU AND QUAD.)
51-60 UNITS PER INCH FOR LONG. IF PLOTTED INSTEAD OF CU AND QUAD.)
61-70 LENGTH OF FREQUENCY AXIS IN INCHES (F10.)
71-80 WHEN LOG10 PLCTS HAVE BEEN REQUESTED ABOVE MUST ALL
      REFER TO LOG10, E.G. UNITS PER INCH = 1.00 MEANS 1 DECADE PER INCH

NEXT CARD, IN COLUMN
5 LINDCT = 0 FOR CONSTANT BANDWIDTH, GIVEN BY BANDW BELOW
6-15 STRFRC = 1 FOR EXPONENTIAL BANDWIDTHS (GIVES LOG10 FREQ.
      SCALE IN ALL PLCTS)
      = 0 FOR LINEAR BANDWIDTHS
16-25 BANDW = APPROX. BANDWIDTH FOR FIXED BANDWIDTH SPECTRA,
      IN HERTZ. (MUST INCLUDE A DECIMAL POINT)
30 INDCW = 1 IF FOURIER COEFFICIENTS TO BE HANNED BEFORE
      SPECTRA COMPUTED

```

```

SUBSEQUENT IC MAX = 0 IF NOT
4-5 CHANNEL NUMBER CF A CHANNEL FOR WHICH SPECTRA ARE WANTED
NUMBER IS USUALLY 25, 29-30 TO CROSS SPECTRUM IS HERE CONSIDERED AS CROSS
14-15, 13-20, 24-25, 29-30 TO CROSS SPECTRUM IS HERE CONSIDERED AS CROSS
CHANNELS ARE DESIGNED WITH ITSELF, I.E. THE NUMBER IN 4-5
SPECTRA ARE APPEARED ONLY IF EACH CHANNEL APPEARS IN THE LIST
SPECTRA ARE OTHER CHANNELS SPECTRUM BETWEEN 2 AND CHANNEL 8, THERE
ON THE 2 AND CROSS SPECTRUM BETWEEN 2 AND BGTH 2 AND 8 IN THE LIST
MUST BE A CARD WITH 8 IN COL. 5 AND 2 IN THE LIST
AND ANOTHER CARD WITH 8 IN COL. 5 AND 2 IN THE LIST.

PHASE CORRECTION DECK( INSERT ONLY IF IPHASE NOT ZERO ) INTERPOLATION
PHASE CORRECTIONS APPLIED WILL BE OBTAINED BY LINEAR
BETWEEN VALUES LESS THAN 6 ) FREQUENCIES ( FIG. 4 ) AT WHICH PHASE
FIRST K CARDS( 1-10, 11-20, ... ) ARE THE FULL 48 FREQUENCIES ARE SUPPLIED, THEN
CORRECTIONS ARE THE FULL 48 FREQUENCIES ARE SUPPLIED, THEN
IF THE LAST FREQUENCY MUST BE LEFT BLANK.

SUBSEQUENT IC MAX SETS CF K PHASE CORRECTIONS( FIG. 4 ) TO BE APPLIED
1-10, 11-20, ... ) FREQUENCIES. EACH SET CONTAIN THE CORRECTIONS
AT THE GIVEN CHANNELS. AND THE SETS ARE TO BE APPLIED AS THE
SPECTRUM CARDS. IF NO CORRECTIONS ARE TO BE BLANK.
CHANNEL, THE K CARDS FOR IN THIRADIANS. A POSITIVE CORRECTION WILL
CAUSE THE CORRECTED PHASE TO LEAD THE UNCORRECTED ONE.

LAST CARD MUST BE BLANK UNLESS ANOTHER FILE IS TO BE PROCESSED. IN
WHICH CASE A COMPLETE NEW SEQUENCE OF CARDS APPROPRIATE TO THE NEW
FILE SHOULD FOLLOW.

DIMENSION ARTAPE(256), TAPRAY(256), PHI(10,10,32), PHISQ(10,10,32),
1 PHIBL(10,10,32), KCHA(10), KCHB(10,10), ICH(10,10), FREQ(32), NEND(33)
2 BANDW(32), MCHAN(10), INDCH(12), BD(32), ACHNAM(9,10), AUNIT(2,10),
3 CAL(32), DC(10,10), DCSQ(10,10), CCBL(10,10), PHASE(32),
4 POS(32), AR2(10,10), AR3(10,10), AM1(10,10), AM2(10,10), AM3(10,10),
5 BR1(10,10), BR2(10,10), BR3(10,10), BM1(10,10), BM2(10,10), BM3(10,10),
6 COSP(10,10), GTIT(18), PCSU(32), NLOGRW(5)
7 EQUIVALENCE (TAPRAY(100), ICH(1,1)), (AKTAP(1), Z(1)), (Z(1), IDUSER)
EQUIVALENCE (TAPRAY(100), ICH(1,1)), (AKTAP(1), Z(1)), (Z(1), IDUSER)
1 (Z(2), MBBLOCK), (Z(3), IBLOCK), (Z(4), NTAP), (Z(5), KCHAN),
2 (Z(6), SAMERC), (Z(11), MCFAN(1)), (Z(31), ACHNAM(1,1))

```



```

39  C J = 1, LC
40  IF (KCHA(KA).GT.ICH(KA,J)) GC TC 39
    KB = KB+1
    KCHB(KA,KB) = ICH(KA,J)
    CONTINUE
    IF (IPHASE.EQ.0) GC TO 41
    CALL IPHASES(0,Z,CO,QU,IFB,FREQ,NEND,BANDW,PHASE,KCA,KCB,FUNDFR,
1    ICMAX,MCHAN,MAXIF,RAD,KCHA)
41  KSKIP = NFILE - IFILE
    IF (KSKIP.GT.0) CALL SKPFL (KSKIP,3)
    IFILE = NFILE
    READ (3,END=1108) ARTAPE
    KBEGIN = 0
    IF (ICTEST1 - ICUSER) 4900,54,4900
    IF (IBSTAR - MBLCK) 55,65,55
54  DO 56 N = 1, N1AP
55  READ (3,END=1108) TAPRAY
56  READ (3,END=1108) ARTAPE
    GC TC 54
    IB = (IBSTAR + IBMAX-1) - MBLCK
    IF (IB) 47CC,60,60
59  IBMAX = IB
60  IBSTAR = MBLCK
    C PRINT IS = 0
65  WRITE (6,62) IDUSER,IFILE,ICMAX,IBMAX,IBSTAR
    IF (IFANN.GT.0) INDOV = 0
    C EXCHANGE 67 K = 1, 10
    DO 67 K = 1, 10
    IT = KCHA(K)
    KCIS = K - IS
    KCHA(KIS) = 0
    IF (KC.LE.0) GC TO 659
    KCHA(KIS) = INCCH(KC)
    IF (INDCH(KC).EQ.0) IS = IS+1
    DO 66 I = 1, 10
659  ICIT = KCHB(K,I)
    KIT = I - IT
    KCHB(KIS,KIT) = 0
    IF (IC.LE.0) GC TO 66
    KCHB(KIS,KIT) = INCCH(IC)
    IF (INDCH(IC).EQ.0) IT = IT+1
    CONTINUE
66  NHAR = 256 / ((2*KCHAN)+1)
67

```



```

77      PAND = BAND+C.50
      I = I+1
      BAND = BAND*1.33352
      NEND(I) = BAND
      MAXIF = I
      IF(NEND(I).LT.JBLOCK) GO TO 77
      NEND(I) = JBLOCK
      DO 78 I = 2,MAXIF
      J = I-1
      JEAND = NEND(I)-NEND(J)
      JEAND*(I) = BAND*FUNDJR
      BAND = NEND(I)
      PCSU(I) = (BAND+C.50)*FUNDJR
      PCSU(J) = PCSU(I)
      FREQ(I) = SQRT(PCSL(I)*PCSU(I))
      FREQ(J) = FREQ(I)
      CEST SET TC READ SOME DATA
      KBLOCK = I
      IF(I.AN.GT.C.AND.NEND(MAXIF).EC.JBLOCK) NEND(MAXIF) = JBLOCK - 1
      ISTART = IFR = 1,MAXIF
      DO 99 KA = 1,10
      DO 98 KB = 1,10
      CCSQ(KA,KB) = 0.0
      LUBL(KA,KB,IFR) = 0.0
      PHISC(KA,KB,IFR) = 0.0
      PHIBL(KA,KB,IFR) = 0.0
      CCNTINUE
      BL SQ = C.0
      C BEGINNING OF BLOCK LGQP
      180  BLOCK = ABLCK
      DO 181 KBC = 1,10
      AR1(KBC) = C.0
      AR2(KBC) = C.0
      AM1(KBC) = C.0
      AM2(KBC) = C.0
      AM3(KBC) = C.0
      CCUSP(KAC,KBC) = 1.10
      CCUSP(KAC,KBC) = 0.0
      BR1(KAC,KBC) = 0.0
      BR2(KAC,KBC) = 0.0

```

```

181 PR3(KBC,KAC) = 0.0
    PM1(KBC,KAC) = 0.0
    PM2(KAC,KBC) = 0.0
    PM3(KAC,KBC) = 0.0
    CONTINUE + BLOCK
182 PLSQ = PLSQ + (BLOCK*BLOCK)
    IFRM = 1
185 CCICCNT = -1
    NCAR = 3,END=1108) TAPRAY
    READ(3,END=1,NTAP)
190 IF(IFR.GT.MAXIF) GC TC 1000
    IFRM = IFRM + 1
19005 IFRM2 = IFRM - INDCW
    IF(IIFR2.EQ.0) GC TC 191
    IF(IIFR2.LT.NSTARW) GC TC 390
    IF(IIFR2.LE.NEND(IFR)) GC TC 191
    CC85C KA = 1,ICMAX
    KCA = KCHA(KA)
    IF(KCA.LE.0) GC TC 850
    CC85C KCB = 1,ICMAX
    KCB = KCB(KA,KB)
    IF(KCB.LE.0) GC TC 850
    PHI(KCA,KCB,IFR) = CCSP(KA,KCB,IFR)
    PHIBL(KCA,KCB,IFR) = CCSP(KA,KCB,IFR)
    PHISC(KCA,KCB,IFR) = CCSP(KA,KCB,IFR)
    CCSP(KA,KCB,IFR) = 0.0
    IF(KCA.EC.KCB) GC TC 850
    PHI(KCB,KCA,IFR) = CCSP(KA,KCB,IFR)
    PHIBL(KCB,KCA,IFR) = CCSP(KA,KCB,IFR)
    PHISC(KCB,KCA,IFR) = CCSP(KA,KCB,IFR)
    CCNT INUEFR+1
    IF(IFR.GT.MAXIF) GC TC 1000
    ISUB = (IFR-1)*(1+(2*KCHAN))
    THARM = IFRM
    IF(NC, (THARM.EQ.TAPRAY(ISUB+1))) GO TO 4800
195 DO 98C KA = 1,ICMAX
    KCA = KCHA(KA)
    IF(KCA.LE.0) GC TC 98C
    IS = ISUB + (2*KCA)
    IF(INC,GT.0) GC TC 196
    ANL = TAPRAY(IS)
    GC TC 199

```

```

196 AR3(KA) = AR2(KA)
    AR1(KA) = TAPRAY(15)
    AM3(KA) = AM2(KA)
    AM2(KA) = AM1(KA)
    AM1(KA) = TAPRAY(15+1)
    IF(IIFARM2.EQ.C) GC TC 197
    IF(IIFARM2.LT.NSTART) GC TC 199
    ARG = CCRNB*((2.0*AR2(KA))-AR1(KA)-AM3(KA))
    GC TC 199
    ARG = 2.0*CCRAH*(AR2(KA)-AR1(KA))
    ARG = 0.0
    CONTINUE
    KCB = KCB(KA,KP) = 1,ICMAX
    IF(KCB.LE.C) GC TC 97C
    IT = ISUB+(2*KCB)
    IF(INDW.GT.J) GC TC 206
    BR1 = TAPRAY(IT)
    BR2 = TAPRAY(IT+1)
    GC TC 205
    BR3(KA,KB) = BR2(KA,KB)
    BR2(KA,KB) = BR1(KA,KB)
    BR1(KA,KB) = TAPRAY(IT)
    BR3(KA,KB) = BR2(KA,KB)
    BR2(KA,KB) = BR1(KA,KB)
    BR1(KA,KB) = TAPRAY(IT+1)
    IF(IIFARM2.EQ.C) GC TC 207
    IF(IIFARM2.LT.NSTART) GC TC 209
    BR1 = CCRNB*((2.0*BR2(KA,KB))-BR1(KA,KB)-BR3(KA,KB))
    BR2 = CCRNB*((2.0*BR2(KA,KB))-BR1(KA,KB)-BR3(KA,KB))
    GC TC 209
    BR1 = 2.0*CCRAH*(BR2(KA,KB)-BR1(KA,KB))
    BR2 = 0.0
    CONTINUE
    CC = ((ARL*BR1)+(AMG*BR2))/2.0
    CC = ((BR1*AMG)-(ARL*BR2))/2.0
    C STCR
    IF(IIFARM2.NE.G) GC TC 801
    IF(KCA.EQ.KCB) GC TC 250
    DC(KCA,KCB) = ARL*BR1+CC(KCA,KCB)
    DC(KCA,KCB) = ARL*BR2+CC(KCA,KCB)
    DCBL(KCA,KCB) = ARL*BR1*BLCK + CCBL(KCA,KCB)
    GC TC 250
    DC(KCA,KCB) = ARL+CC(KCA,KCB)
    DC(KCA,KCB) = ARL*ARL+CC(KCA,KCB)
    DCBL(KCA,KCB) = (ARL*BLCK) + CCBL(KCA,KCB)
250

```

[illegible]

```

KCB = KCB(KA,KB)
IF (KCB.LT.0) GC TC 2500
SUMCC = C.CC
IF (KCB.NE.0) 1200 1150,1200
C WRITE HEADING(6,1150) IDUSER,MCHAN(KCA), (ACHNAM(L,KCA),L=1,9)
1150 WRITE(6,1150) IDUSER,MCHAN(KCA), (ACHNAM(L,KCA),L=1,9)
IF (IACCM.GT.0) WRITE(6,14)
IF (IACCM.GT.0) WRITE(6,14)
IF (IACCM.GT.0) WRITE(6,14)
1 KCL=1,2,3
1 GC TC 1200 1150,1200
C WRITE HEADING(6,1200) IDUSER,MCHAN(KCA), (ACHNAM(L,KCA),L=1,9),MCHAN(KCB),
1200 2 (ACHNAM(L,KCB),L=1,9),MCHAN(KCB),
WRITE(6,1150) IDUSER,MCHAN(KCA), (ACHNAM(L,KCA),L=1,9),MCHAN(KCB),
IF (IACCM.GT.0) WRITE(6,14)
IF (IACCM.GT.0) WRITE(6,14)
1 (AUNITIS(KL,KCB),KL=1,2), (AUNITIS(KL,KCA),KL=1,2),
2 (AUNITIS(KL,KCB),KL=1,2), (AUNITIS(KL,KCA),KL=1,2),
3 (AUNITIS(KL,KCB),KL=1,2), (AUNITIS(KL,KCA),KL=1,2),
C NGW SET 1500 DENSITY, STD.DEV., REGRESSION COEFF
1209 VIC = BLANK*VID
CIV FOR REAL PART
C C = PH(KCA,IFB)/(VID*VIC) - (BLMAX*CO*CC)/(BLMAX-1.)
FCSQ = ((PH(KCA,IFB)/(VID*VIC) - (BLMAX*CO*CC)/(BLMAX-1.))
CCSQ = ((PH(KCA,IFB)/(VID*VIC) - (BLMAX*CO*CC)/(BLMAX-1.))
A(IIFB) = CC
AD(IIFB) = CC - (CCSQ/BLRCC)
AE(IIFB) = CC - (CCSQ/BLRCC)
IF (IIFB) = AE(IIFB)
AD(IIFB) = AE(IIFB)
AE(IIFB) = AE(IIFB)
CCNT = 1
CUP = ((PH(KCA,IFB)/(VID*VIC) - (BLMAX*CO*CC)/(BLMAX-1.))
IF THIS WAS EC,KKCB) GC TC 1300
C IF NCT, = PH(KCB,KCA,IFB)/DIV
C QUSQ = ((PH(KCB,KCA,IFB)/(VID*VIC) - (BLMAX*CO*QU))/(BLMAX-1.))

```

```

CURSQ = (PHI*(KCB,KCA,IFB)/VID) - (BL*CU)/((BL*BL)/BLMAX))
ARL = (PHI*(KCA,KCA,IFB)/DIV
RRL = (PHI*(KCB,KCB,IFB)/DIV
CU = CC*CU
FAD = ARL*CU
ARL = ARL*BL
CCRCFF = ARL*AD/SORT(ARL)
IF( I PHASES( 2, 2, CC, QU, IFB, FREQ, NEND, BANDW, PHASE, KCA, KCB, FUNDFR,
1 ICMAX, NCHAN, MAXIF, RAD, KCHA)
ARG = QU/CC
FCC = CC*FREQ(IFB)
A(IFB) = CC
AE(IFB) = CC + (CCSC/PLRCCT)
BE(IFB) = CU
RE(IFB) = CU + (GUSC/BLRCCT)
IF( I PLCT, 4) GC TC 12092
A(IFB) = A(IFB)*FREQ(IFB)
AE(IFB) = AE(IFB)*FREQ(IFB)
BE(IFB) = BE(IFB)*FREQ(IFB)
REC(ITC) 12092
A(IFB) = CCRCFF
AE(IFB) = CCRCFF
BE(IFB) = BMG
CONTINUE
IF( ARG, LG, 10000) GC TC 1210
BMG TC 1212
BMG = 57.296 * ATAN2(QU,CC)
C WRITE OUT CRCS SPECTRA
1212 1 CRCS SPECTRA
1300 1 CRCS SPECTRA
1499 1 CRCS SPECTRA
SUMCC = SUMCC + (CD*VID)

```

```

1500 SUMCL = SUMCL + (CU*VIC)
CONTINUE
IF (KCA, KCB) GC TC 1501
WRITE(6,1535) SUMCG, (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
1 SUMCL = (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
GC TC 1502
WRITE(6,1545) SUMCG, (AUNITS(KL,KCA), KL=1,2)
1501 CONTINUE
IF (KCA, KCB) GC TC 1503
DCDIF = CC(KCA, KCB) - (DC(KCA, KCA)*DC(KCB, KCB))
1502 DCDIF = CC(KCA, KCB) - (DC(KCA, KCA)*DC(KCB, KCB))
WRITE(6,1525) DC(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2), DCSC(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2), DCBL(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2), DCIF, (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
2 1,2), DCIF, (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
3 GC TC 1504
WRITE(6,1526) DC(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), DCSC(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), DCBL(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), DCIF, (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
1503 CONTINUE
1 (AUNITS(KL,KCA), KL=1,2), DCBL(KCA, KCB), (AUNITS(KL,KCA), KL=1,2), DCIF, (AUNITS(KL,KCA), KL=1,2), (AUNITS(KL,KCB), KL=1,2)
1504 IF (IPLCT.EC.0) GC TC 2500
LPL TC 1740,1750,1750,1740,1740,1750,1750,IPLCT
GC IF (INOCY) 1770,1770,1750
1740 IF (PCS(1).NE.0.0) GC TC 17505
1750 FOSL(1) = PCS(1)+PCS(2)-PCS(1)
FOSU(1) = PCS(1)+PCS(2)
PCS(1) = PCS(2)
GC TC 1770
IF (IFRSL.GT.0) GC TC 1770
DCSU(1) = ALCCG(1,MAXIF)
IF (PCS(1).LE.0.0) PCSU(1) = 0.001*POS(1)
IF (PCS(1).LE.0.0) PCSU(1) = 0.001*POS(1)
POS(1) = ALCCG(1,MAXIF)
POS(1) = ALCCG(1,MAXIF)
GC TC 170015
IF (IFLCT.LE.0.0) GC TC 70015
IF (KCA.NE.KCB) GC TC 7100
IF (DO 7003) IL82CC,8100,8110
8100 IF (A(1).LE.0.0) AE(1) = A(1)
8110 IF (A(1).LE.0.0) AE(1) = A(1)
GC TC 8300
AE(1) = -AE(1)
AD(1) = -AD(1)
IF (AD(1).LE.0.0) AD(1) = A(1)
AD(1) = ALCCG(1,MAXIF)
AD(1) = ALCCG(1,MAXIF)
AE(1) = ALCCG(1,MAXIF)
IF (KCA.EC.KCB) GC TC 7003

```



```

7112 DO 7113 IC = 1,9
      ADDX = IC-1
      XS = 2.0C+(C.48*ADDX)
      CALL SYMBCL(XS,YS,C.14,ACHNAM(,C,KC),O.OO,4)
      CONTINUE
      IF (KC.EQ.KCB) GC TO 7120
      KCB = KC
      ACXY = O.25
      GC TC 7112
7120 IF (IFRSCLE.GT.O) GC TC 7122
      IFRSCLE=1
      IF (IAXIS.GT.O) GC TC 7121
      CALL SCALE(PCS,MAXIF,4.OO,XMIN3,CX3,1)
      DO 71215 KIF=1,MAXIF
      IF (IAXIS.NE.O) POS(KIF) = (PCS(KIF)-XMIN3)/DX3
      POSU(KIF) = (PCSU(KIF)-XMIN3)/DX3
      IF (PCS(KIF).GT.FRAXLN) POS(KIF) = FRAXLN
      IF (PCSU(KIF).GT.FRAXLN) POSU(KIF) = FRAXLN
      IF (PCS(KIF).LT.O.O) PCS(KIF) = O.O
      IF (PCSU(KIF).LT.O.O) PCSU(KIF) = FRAXLN
      IF (PCSEL(KIF).GT.FRAXLN) POSL(KIF) = FRAXLN
      IF (PCSEL(KIF).LT.O.O) PCSEL(KIF) = O.O
      CONTINUE
71215 IF (LIACCT.EQ.1) GO TO 71225
7122 GO TC 7130,71225,71225,7130,71225,71225
      CALL AXIS(2.OO,3.5O,15HLOG10 FREQUENCY,-15,FRAXLN,O.O,XMIN3,DX3)
      GC TC 7140
7130 CALL AXIS(2.OO,3.5O,9HFREQUENCY,-9,FRAXLN,O.O,XMIN3,DX3)
7140 IF (KCA.EQ.KCB) GC TO 7130
      IF (LPL.GT.1) GO TO 7170
      GC TC 7150,7145,7145,7142,7155,7155,7155
      CALL AXIS(2.OO,3.5O,13H FREQ X CUSP ,+19,5.OO,9O.O,XMIN1,
1 DX1)
7142 1 GC TC 718C
7145 CONTINUE
      CALL AXIS(2.OO,3.5O,17HLCG10 CC-SPECTRUM,+17,5.OO,9C.O,XMIN1,
1 DX1)
7150 GC TC 718C
      CALL AXIS(2.OO,3.5O,11HCC-SPECTRUM,+11,5.OO,9O.O,XMIN1,DX1)
7155 GC TC 718C
      CALL AXIS(2.OO,3.5O,9HCCPERENCE,+9,5.OO,9C.O,XMIN1,DX1)
7170 GC TC 718C
      CALL AXIS(7175,7171,7171,7175,7172,7172,7172
1 DX2)
71715 1 GC TC 7180
      CALL AXIS(2.OO,3.5O,13H FREQ X QUSP ,+19,5.OO,9O.O,XMIN2 ,
1 DX2)

```

```

7171      CONTINUE
      CALL AXIS(2.00,3.50,15HLCG1C QUAD-SPECTRUM,+15,5.00,90.0,XMIN2,
1 DX2)
      GC TC 7180
7175      CALL AXIS(2.00,3.50,13HQUAD-SPECTRUM,+13,5.00,90.0,XMIN2,DX2)
      GC TC 7180
7172      CALL AXIS(2.00,3.50,5HPHASE,+5,5.00,90.0,XMIN2,DX2)
7180      IF(IPLCT.EQ.4) GC TC 7181
      CALL LABEL(AUNITS(1,KCA),AUNITS(1,KCB),2)
7181      CONTINUE
      IF (LPL.GT.1) GC TC 7190
      CALL FLVAL(A,AD,AE,PCS,PCSU,POSL,MAXIF)
      LPL = 2
      CALL FLCT (C.,YREGRG,-3)
      GC TC 7110
7190      CALL FLVAL(R,ED,BE,PCS,FCSU,PCSL,MAXIF)
      LPL = 1
      GC TC 2800
7300      GC TC (7310,7305,7302,7310,7305,7302
7302      CALL AXIS(2.00,3.50,
14H FREQ X SPECT
, +20,5.00,90.0,XMIN1,
DX1)
      GC TC 7320
7305      CONTINUE
      CALL AXIS(2.00,3.50,14HLCG1C SPECTRUM,+14,5.00,90.0,XMIN1,DX1,
GC TC 7320)
7310      CALL AXIS(2.00,3.50,8HSPECTRUM,+8,5.00,90.0,XMIN1,DX1)
7320      IF(IPLCT.EQ.4) GC TC 7327
      CALL LABEL(AUNITS(1,KCA),AUNITS(1,KCB),1)
7327      CALL FLVAL(A,AD,AE,PCS,PCSU,POSL,MAXIF)
2800      CALL FLCT (C.,YREGRG,-3)
2900      CONTINUE
3000      CONTINUE
      GC TC 5000
4700      WRITE(6,4705) MBLUCK,IBSTAR,IBMAX,IFILE
      GC TC 5000
4800      WRITE(6,4805) IHARM,TAFRAY(ISUB+1),NT,IH,KCHAN
      GC TC 5000
4900      WRITE(6,4905) IFILE,IDUSER,ICTEST
5000      GC TC 1
5020      IF (LCPT.LE.C) GC TC 5015
      CALL PLCTE
      REH INCT
5015      CALL EXIT
      FCFMAT(17F10.0)
5      FCFMAT(1844)
      FCFMAT(45HC FCLRIER CCEFFICIENTS WERE HANNED BY FTRC /)
14      FCFMAT(19,1X,6(I5,5X))
15      FCFMAT(15,2F10.0,15)
16

```



5005 FORMAT (37F6 RUN TERMINATED IN AN ORDERLY MANNER )  
 CONTINUE  
 END

JOHN GARRETT

SUBROUTINE FLVAL(A,AD,AE,PCS,PCSU,PCSL,NVAL)  
 C SPLCT = FLVAL AND LABEL  
 C AUGUST 6 1968  
 DIMENSION A(1),AE(1),AD(1),PCS(1),PCSU(1),PCSL(1)  
 DO 20 K = 1,NVAL  
 Y = A(K) + 3.50  
 XP = PCS(K) + 2.00  
 CALL FLCT(XP,Y,+3)  
 XP = PCSU(K) + 2.00  
 CALL FLCT(XP,Y,+2)  
 Y = AD(K) + 3.50  
 CALL FLCT(X,Y,+3)  
 CALL TIC(X,Y)  
 Y = AE(K) + 3.50  
 CALL FLCT(X,Y,+2)  
 CALL TIC(X,Y)  
 CCNTINCE  
 RETURN  
 END

20

51

SUBROUTINE TIC(X,Y)  
 X = X+0.02  
 CALL FLCT(X,Y,+2)  
 X = X-0.04  
 CALL FLCT(X,Y,+2)  
 X = X+0.02  
 CALL FLCT(X,Y,+2)  
 RETURN  
 END

JOHN GARRETT

C SUBROUTINE LABEL (U1,U2,ISY)  
C VERSION OF APRIL 26 1958

DIMENSION U1(2),U2(2)

X = 1.70

Y = 7.30

HT = 0.14

CALL SYMBCL(X,Y,HT,1H,90.0,1)

Y = Y+0.12

CALL SYMBCL(X,Y,HT,U1,90.0,8)

Y = Y+1.00

GC TC (10,20), ISY

CALL SYMBCL(X,Y,HT,1HX,90.0,1)

Y = Y+0.20

CALL SYMBCL(X,Y,HT,U2,90.0,8)

Y = Y+0.96

GC TC 30

X = 1.70

XSUF = 1.0

CALL SYMBCL(XSUP,Y,HT,1H2,90.0,1)

Y = Y+0.18

CALL SYMBCL(X,Y,HT,7H/HERTZ),90.0,7)

RETURN

END

20

10

30

SUBROUTINE PHASES(N,A,CC,QU,IFB,F,ND,8W,PHASE,KCA,KCB,FFR,ICM,  
1 MCHAN, MAXIF, RAD, KCHA)  
DIMENSION A(1),F(1),NC(1),BW(1),MCHAN(1),PHASE(50,1),PHCOR(50,1),  
1 SUMF(12), MWR(10), KCHA(1)  
KTRL = 10,100,500, KTRL

GC TC (10,100,500), KTRL

CC 9800 NZERC = 1.48

CC 9800 NZERC = 1.11

PHCOR( NZERC,NZERO ) = 0.0

CONTINUE

MPRE = 0

CC 9820 KARC = 1.6

READ( 5,9825 ) ( PHASE(JFRE,1), JFRE = 1,8 )

CC 9820 LFRE = 1.8

MPRE = MPRE + 1.8

FIN = PHASE( LFRE,1 )

IF( FIN.EC.C ) GC TC 9830

PHCOR( MPRE,1 ) = FIN

CONTINUE

KFR = 48

10

9800

9820





```

C
C
C
C
C
C
END
THIS SUBROUTINE WHEN CALLED MOVES THE TAPE CN LGICAL UNIT NUNIT
FAST KSKIP END OF FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN
FORTRAN BINARY. IF KSKIP IS ZER0 OR NEGATIVE THE ROUTINE RETURNS
WITHOUT MOVING THE TAPE.

```

```

SUBROUTINE SKFFL (KSKIP,NUNIT)
IF (KSKIP.LT.1) RETURN
CC 2C I=1,KSKIP
1C READ (NUNIT,END=20) ICUMPY
2C CC TC 10
CC CONTINUE
CC RETURN
END

```

```

SUBROUTINE SCALE (X,N,S,YMIN,DY,K)
DIMENSION X(2)
LGICAL FLAG
FLAG = .FALSE.
YMAX=X(1)
YMIN=YMAX
NP=N*K
CC 1C I=1,NP,K
IF (YMAX-X(I)) 5,6,6
5 YMAX=X(I)
6 IF (X(I)-YMIN) 7,10,10
7 YMIN=X(I)
10 CC CONTINUE
105 IF (RANGE=YMAX-YMIN) GO TC 11
IF (RANGE.NE.0.0) GO TC 11
DY = C.0
CC 12 I=1,NP,K
12 X(I) = 0.0
11 SPFACT = RANGE / (S )
SFLOGG = ALCG10(SFACT)
IF (SFLOGG.GE.0.0) GO TC 13
13 SFLOGG = SFLOGG - 1.
ISFLCG = SFLOGG
ISFACT = SFLOGG / 10.** ISFLCG

```

```

SCAL00020
SCAL00030
SCAL00034
SCAL00036
SCAL00040
SCAL00050
SCAL00060
SCAL00070
SCAL00080
SCAL00090
SCAL00100
SCAL00110
SCAL00120
SCAL00130
SCAL00140
SCAL00150
SCAL00160
SCAL00170
SCAL00180
SCAL00190
SCAL00200
SCAL00210
SCAL00220
SCAL00230
SCAL00240

```

SCALC0256C  
SCALC0270C  
SCALC0286C  
SCALC0290C  
SCALC0310C  
SCALC0320C  
SCALC0330C  
SCALC0340C  
SCALC0350C  
SCALC0360C  
SCALC0370C  
SCALC0380C  
SCALC0390C  
SCALC0400C  
SCALC0415C  
SCALC0420C  
SCALC0430C  
SCALC0444C  
SCALC0446C  
SCALC0447C  
SCALC0448C  
SCALC0450C  
SCALC0460C  
SCALC0470C  
SCALC0480C

```

IF(SFACT.GT.1.0) GC TC 14
CY = 10.0 ISFLCG
14 IF(SFACT.GT.2.0) GC TC 16
CY = 2.0 ISFLCG
16 IF(SFACT.GT.4.0) GC TC 17
CY = 4.0 ISFLCG
17 IF(SFACT.GT.5.0) GC TC 18
CY = 5.0 ISFLCG
18 IF(SFACT.GT.8.0) GC TC 19
CY = 8.0 ISFLCG
19 LY = 10.0 ISFLCG + 1)
TEMP = YMIN / DY
IF(FLAG) GC TC 21
IF(TEMP.LT.0) TEMP = TEMP - 1.0
YMIN = TEMP
FLAG = FLAG + 1
TEMP = YMIN + S * DY
SCALE = 1.0 / TEMP
IF(YMAX.GT. TEMP) GC TC 105
DO 20 I=1,NP,K
X(I)=(X(I)-YMIN)/DY
20 RETURN
END
C
21

```

```

SUBROUTINE AXIS(X,Y,PCD,NC,SIZE,THETA,YMIN,DY)
DIMENSION BCD( 2 )
SIG = 1.0
IF( NC ) 1 , 2 , 2
1 SIG = -1.0
NAC = IABS( NC )
TH = THETA * 0.017+53254
N = N
NTH = CCS( TH )
STH = SIN( TH )
SCALE = DY
ADY = YMIN
ABSU = 0.0
EXP
C****

```

```

9      IF( ADY , 9 , 16 , 9
12     IF( ABS( ADY ) - 100.0 ) 10 , 12 , 12
        ADY = ADY * 0.1
        ABSU = ABSU * 0.1
        EXP = EXP + 1.0
        GC TC
14     ADY = ADY * 10.0
        ABSU = ABSU * 10.0
        EXP = EXP - 1.0
16     IF( ABS( ADY ) - 0.01 ) 14 , 16 , 16
C$**** INITIALIZE THE MAIN LCCP *****
C      XL , YN , YL ETC. ARE FOR THE LINERS
C      XT , YT ETC. ARE FOR THE TITLE
C$****
18     XS = X
        YS = Y
        TEMP = 0.2 * SIG - 0.05
        XNA = XS - TEMP * CTH - 0.0857 * CTH
        YNA = YS + TEMP * CTH - 0.0857 * CTH
        NAFT = ABS( ABSU ) * 100.0 * C.5
        ICK = MOD( ICK , 10 ) .EC. 0 ) NAFT = 2
        IF( MOD( ICK , 100 ) .EC. C ) NAFT = 1
        CALL NUMBER( XNA , YNA , 0.10 , ABSU , THETA , NAFT )
        XLB = XS + IN * CTH
        YLB = YS + IN * CTH
        TEMP = 0.1 * SIG
        XLA = XLB - TEMP * CTH
        YLA = YLB + TEMP * CTH
        ABY2 = N/2
        THIS IS THE MAIN LCCP *****
C$**** DO 20 I = 1 , N
C$**** NEXT I THE NUMBER *****
        XNA = XNA + CTH
        YNA = YNA + CTH
        ABSU = ABSU + ADY
        NAFT = 3
        ICK = ABS( ABSU ) * 100.0 * C.5
        IF( MOD( ICK , 10 ) .EC. 0 ) NAFT = 2
        IF( MOD( ICK , 100 ) .EC. C ) NAFT = 1
        CALL NUMBER( XNA , YNA , 0.10 , ABSU , THETA , NAFT )
        CALL NCH PERHAPS THE TITLE *****
C$**** IF( I .NE. 7
        INC = 7
        TEMP = SIZE * 0.5 - 0.06 * INC
        TEMPB = 1 - 0.07 * SIG * C.30
        XT = XS + TEMP * CTH - TEMPB * CTH

```

```

15      YI = YS + TEMP*STH + TEMP*CTH
16      CALL SYMCL( XT, YI, GO, TIC, 15 )
17      TEMP = ( TAC*EC - 6.0 ) * C.12
18      YI = XT + TEMP*CTH
19      CALL SYMCL( XT, YI, C.14, 7H(XIC ), THETA, 7 )
20      XT = XT + C.48*CTH - C.07*STH
21      YI = YI + C.48*STH + C.07*CTH
22      CALL NUMBER( XT, YI, C.10, EXP, THETA, -1 )
23      CONTINUE
24      C*****
25      TRACE THE LINE BACKWARDS TO THE ORIGIN *****
26      CALL PLCT( XLA, YLA, +3 )
27      CALL PLCT( XLR, YLR, +2 )
28      DO 44 I = 1, N, TIC *****
29      BOTTOM = XLR - STH
30      XLR = YLR - STH
31      CALL PLCT( XLR, YLR, +2 )
32      TOP = PLCT( XLR, YLR, +2 )
33      XLA = XLR - STH
34      CALL PLCT( XLA, YLA, +2 )
35      CALL PLCT( XLR, YLR, +2 )
36      CONTINUE
37      RETURN
38      END

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

PRINTER PLOT OF AMPLITUDE FROM FOURIER COEFFICIENT TAPE

THE FIRST INPUT CARD SHOULD CONTAIN UP TO 72 CHARACTERS OF PLOT IDENTIFICATION FOR USE BY THE COMPUTING CENTER STAFF AND IS REPRODUCED ON THE PLOTTER OUTPUT. THIS CARD MUST BE PRESENT WHETHER PLOTS ARE PRODUCED OR NOT. IT IS PRESENT ONLY ONCE IN THE DECK. THE FOLLOWING SET OF CARDS IS PRESENT FOR EACH FILE OF FOURIER COEFFICIENTS PROCESSED.

CONTROL CARD DECK MAKEUP

1) KFILE,KCHAN,KSTART,KSTOP,KPLOT,KRULE,KLIST,KPUNCH,LOGLIN,LOGLOG (1015)

WHERE,

KFILE = NO. OF FILE ON FOURIER COEFFICIENT TAPE

KCHAN = NO. OF FOURIER COEFFICIENT TAPE CHANNEL DESIRED

KSTART = FIRST BLOCK TO BE INCLUDED IN ANALYSIS

KSTOP = LAST BLOCK TO BE INCLUDED IN ANALYSIS

KPLOT = 1 FOR CALCOMP PLOT  
= 0 FOR NO CALCOMP PLOT

KRULE = 1 FOR A COLUMN OF PERIODS PER INCH ON THE PLOT  
= 0 IF THIS IS NOT DESIRED

KLIST = 1 FOR A PRINTER PLOT  
= 0 TO SUPPRESS THE PRINTER PLOT

KPUNCH = 1 TO PUNCH A BINARY DECK OF THE INTERMEDIATE SUMS  
= 0 OTHERWISE

LOGLIN = 1 TO OBTAIN A LOG VS LINEAR SPECTRUM PLOT  
= 0 OTHERWISE

LOGLOG = 1 TO OBTAIN A LOG LOG SPECTRUM PLOT  
= 0 OTHERWISE

2) TITLE CARD WITH UP TO 72 COLUMNS OF IDENTIFICATION FOR PRINTER OUTPUT AND PLOTS

THE FOLLOWING CARD IS PRESENT ONLY IF LOGLIN = 1 ON CARD 1)

3) ADCDE,NPDCDE,FSTART,DPINCH (2110,2F10.0)

WHERE,

NDCDE = NO. OF FREQUENCY DECADES TO BE PLOTTED

NPOCDE = MAXIMUM DESIRED NO. OF SPECTRAL ESTIMATES PER DECADE

FSTART = ANOTATION TO APPEAR ON FIRST FREQUENCY DECADE ON PLOT

DPINCH = NO. OF FREQUENCY DECADES TO BE PLOTTED PER INCH

THE FOLLOWING CARD IS PRESENT ONLY IF LOGLOG = 1 ON CARD 1)

4) NDCDE,NPOCDE,FSTART,DPINCH,NYDCDE,CY (2I10,2F10.0,I10,F10.0)

WHERE,

NDCDE = NO. OF FREQUENCY DECADES TO BE PLOTTED

NPOCDE = MAXIMUM DESIRED NO. OF SPECTRAL ESTIMATES PER DECADE

FSTART = ANOTATION TO APPEAR ON FIRST FREQUENCY DECADE ON PLOT

DPINCH = NO. OF FREQUENCY DECADES TO BE PLOTTED PER INCH

NYDCDE = NO. OF DECADES TO BE PLOTTED ON SPECTRAL DENSITY AXIS

DY = NO. OF DECADES PER INCH TO BE PLOTTED ON SPECTRAL DENSITY AXIS

INPUT TAPE IS ON LOGICAL UNIT 9

A BLANK CARD WILL TERMINATE THE RUN OR ANOTHER COMPLETE SET OF CARDS WILL DO A SECOND ANALYSIS

THE PROGRAM DOES AN ANALYSIS ON THE FOURIER COEFFICIENTS DEFINED BY THE FIRST CONTROL CARD. THE FOURIER COEFFICIENT NO., ITS FREQUENCY, ITS MEAN AMPLITUDE AND ITS 95 PERCENT CONFIDENCE INTERVAL ARE PRINTED. THE MEAN AND 95 PERCENT CONFIDENCE INTERVAL ARE PLOTTED ON THE PRINTER. ONLY THE MEAN IS PLOTTED ON THE CALCOMP PLOTTER.

THE PROGRAM WILL HANDLE ONLY ONE CHANNEL AT A TIME FROM THE FOURIER COEFFICIENT TAPE











66



[illegible]

[illegible]

CF DATA PCINTS IN EACH FCURIER TRANSFORM  
 CALLING SEQUENCE  
 CALL FCINPT (NUNIT, IDUSER, KBLOCK, NSAMPL, JRCHAN, LCHN, DFREQ, A, IND)  
 WHERE,  
 NUNIT = NO. OF TAPE UNIT CONTAINING THE COCEAN TAPE  
 IDUSER = 9 DIGIT USER IDENTIFICATION NUMBER (RETURNED)  
 KBLCKK = CONSECUTIVE NUMBER OF BLOCK OF COEFFICIENTS ON THE  
 COCEAN TAPE (RETURNED)  
 NSAMPL = NO. OF POINTS IN THE FOURIER TRANSFORM (RETURNED)  
 JRCHAN = NO. OF TRANSFORMED CHANNELS TO BE RETURNED  
 LCHN = FIRST LOCATION OF AN ARRAY CONTAINING THE LOCATIONS OF  
 THE JRCHAN CHANNELS TO BE RETURNED  
 DFREQ = FREQUENCY OF DIGITAL DATA TRANSFORMED (RETURNED)  
 JUNITS = 8 CHARACTER NAMES OF UNITS OF DATA (RETURNED)  
 A = ARRAY INTO WHICH COEFFICIENTS ARE TO BE RETURNED  
 IND = 0 FOR NORMAL RETURN  
 = 1 FOR END OF INPUT FILE

IF THE CALLING SEQUENCE  
 CALL FCINPT (-NUNIT, IDUSER, KBLCKK, NSAMPL, JRCHAN, LCHN, DFREQ, A, IND)  
 IS EXECUTED, TITLE INFORMATION ABOUT THE TAPE WILL BE PRINTED AND  
 ALL INFORMATION EXCEPT THE VECTOR A WILL BE RETURNED  
 THE ADDRESS OF THE FIRST COEFFICIENT FOR EACH OF THE JRCHAN  
 TRANSFORMS RETURNED WILL BE GIVEN BY (NSAMPL+2)\*(KCHAN-1) +  
 WHERE KCHAN = NO. OF CHANNEL REQUIRED

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

SUBROUTINE FCINPT (NUNIT, IDUSER, KBLOCK, NSAMPL, JRCHAN, LCHN, DFREQ, JU
1  NITS, A, IND)
DIMENSION MTAPE(255), ATAPE(255), A(1), LCHN(1), JUNIT(1)
EQUIVALENCE (MTAPE(1), ATAPE(1))
KUNIT = IABS(NUNIT)
IF (KUNIT, END=70) MTAPE
READ (KUNIT, 0) GO TO 50
IDUSER = MTAPE(1)
KBLOCK = MTAPE(2)
NSAMPL = MTAPE(3)
DFREQ = ATAPE(6)
IND = 0
READ (KUNIT, END=70) MTAPE
K = 1
DO 30 N=1, NCoeff
  NOLOC = 2*N - 1
  IF (K, LE, NCRSEQ) GO TO 10
  READ (KUNIT, END=70) MTAPE
  K = 1
  NOLOC = K*KINCR + KOFF
  DO 20 J=1, JRCHAN
    JLOC = (J-1)*JINCR + NOLOC
    A(JLOC) = ATAPE(JLOC)
    A(JLOC+1) = ATAPE(JLOC+1)
    K = K + 1
  RETURN = MTAPE(1)
  IDUSER = MTAPE(2)
  KBLOCK = MTAPE(3)
  NSAMPL = ATAPE(6)
  IND = 0
  NCoeff = MTAPE(3)/2 + 1
  JCHANS = MTAPE(5)
  DO 55 I=1, JRCHAN
    J = LCHN(I)
    JUNIT(2*I-1) = MTAPE(2*I+119)
    JUNIT(2*I) = MTAPE(2*I+120)
    KINCR = 2*JCHANS + 1
    KOFF = -KINCR
    NCRSEQ = 256/KINCR
    JINCR = 2*NCoeff
    WRITE (6, 4001) IDUSER, MTAPE(3), JCHANS, DFREQ
    DO 60 J=1, JCHANS
      WRITE (6, 4002) J, MTAPE(J+10)
    WRITE (6, 4003) J, MTAPE(J+10)
    LSTART = LSTART + 22
  LSTOP = LSTART + 8
55
50
10
20
30

```

```

        WRITE (6,4004) (MTAPE(K),K=LSTART, LSTOP)
        WRITE (6,4005) (MTAPE(2#J+119),MTAPE(2#J+120),ATAPE(J+140)
        RETURN
    70 IND = 1
        RETURN
    4001 FORMAT (27HOCCEAN TAPE SPECIFICATIONS/20HOUSER IDENTIFICATION,110
        1/15H TRNG SIZE,16/32H NO. OF CHANNELS TRANSFORMED WAS,13/23H
        2SA. LING FREQUENCY WAS,10.4,15H SAMPLES/SECOND)
    4002 FORMAT (20HCHANNEL INFORMATION/)
    4003 FORMAT (8HCHANNEL,13,24H FROM OCEAN TAPE CHANNEL,13)
    4004 FORMAT (1H,9A4)
    4005 FORMAT (19H UNITS OF DATA ARE ,2A4/24H CALIBRATION FACTOR WAS ,1PE
        116.7)
        END

```

CCCCC

THIS SUBROUTINE DRAWS A HORIZONTAL TIC LINE THE PLOT AT THE POSITION OF THE PEN AT THE TIME OF THE CALL. THE TIC IS 0.04 INCHES LONG. THE PEN IS RETURNED TO ITS POSITION AT THE TIME OF THE CALL.

```

SUBROUTINE HORTIC
CALL WHERE (X,Y)
CALL PLOT (X-0.02,Y,2)
CALL PLOT (X+0.02,Y,2)
CALL PLOT (X,Y,2)
RETURN
END

```

SUBROUTINE REPLCE (M,N)

```

M = N
RETURN
END

```

CCCCC

THIS SUBROUTINE WHEN CALLED MOVES THE TAPE ON LOGICAL UNIT MUNIT PAST KSKIP END OF FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN FORTTRAN BINARY. IF KSKIP IS ZERO OR NEGATIVE THE ROUTINE RETURNS WITHOUT MOVING THE TAPE.





[illegible]



UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

1. REPORT ORIGINATOR (Name and address)		2a. REPORT SECURITY CLASSIFICATION	
Naval Postgraduate School Monterey, California		Unclassified	
3. REPORT TITLE		2b. GROUP	
DIGITAL ANALYSIS OF TURBULENCE DATA ON THE IBM 360/67 AT THE NAVAL POSTGRADUATE SCHOOL			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)			
Technical Report, July 1969			
5. AUTHOR(S) (First name, middle initial, last name)			
J. R. Wilson N. E. J. Boston W. W. Denner			
6. REPORT DATE	7a. TOTAL NO OF PAGES	7b. NO OF REFS	
1 July 1969	79	3	
8. CONTRACT OR GRANT NO		9a. ORIGINATOR'S REPORT NUMBER(s)	
NAVORD CONTRACT ORD TASK-93C-005/ PROJECT NO 551-1/UR104-03-01		NPS-58UW9071A	
		9b. OTHER REPORT NUMBER (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT			
This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
This project funded by: Naval Ordnance Systems Command		Naval Postgraduate School Monterey, California	
13. ABSTRACT			
<p>A system of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.</p> <p>The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FCPLLOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra phase and coherence are also printed out. The program UBC FCPLLOT provides a Cal-comp plot of the spectra for qualitative analysis.</p> <p>These programs have been tested on the IBM 350/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U.B.C. machine.</p> <p>A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.</p>			

DD FORM 1473

1 NOV 65

(PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

REF ID	LINE A		LINE B		LINE C	
	PCLE	BT	PCLE	BT	PCLE	BT
Digital process turbulence Spectral analysis						

DD FORM 1 NOV 68 1473 (BACK)

S/N 0101-807-6821

UNCLASSIFIED

Security Classification

A-31409